MDPI

Article

IoT Assisted Automatic Driver Drowsiness Detection through Facial Movement Analysis using Deep Learning and U-Net based architecture

Shiplu Das [1,†,‡], Sanjoy Pratihar [1,‡], Buddhadeb Pradhan [2], Rutvij H. Jhaveri[3] and Francesco Benedetto [4,*]

[1] Computer Science and Engineering, Indian Institute of Information Technology, Kalyani, 741235, West Bengal, India; shiplud63@gmail.com, sanjoy@iiitkalyani.ac.in
[2] Computer Science and Engineering, University of Engineering and Management, Kolkata, India; buddhadebpradhan@gmail.com
[3] Computer Science and Engineering, School of Technology, Pandit Deendayal Energy University, Gandhinagar, India ; rutvij.jhaveri@sot.pdpu.ac.in
[4] Signal Processing for Telecommunications and Economics, Roma Tre University, Rome, Italy ; francesco.benedetto@uniroma3.it
* Correspondence: francesco.benedetto@uniroma3.it;

Abstract: The main purpose of a detection system is to ascertain the state of an individual's eyes, whether they are open and alert or closed, and then alert them to their level of fatigue. As a result of this, they will refrain from approaching the accident site. In addition, it would be advantageous for folks to be promptly alerted in real-time before the occurrence of any calamitous event affecting everyone. The implementation of Internet-of-Things (IoT) technology in driver action recognition has become imperative due to the ongoing advancements in Artificial Intelligence (AI) and Deep Learning within Advanced Driver Assistance Systems (ADAS), which are significantly transforming the driving encounter. This work presents a deep learning model that utilizes a CNN-Long Short-Term Memory network to detect driver sleepiness. We employ different algorithms on the dataset such as EM-CNN, VGG-16, GoogleNet, AlexNet, ResNet50, and CNN-LSTM. The aforementioned algorithms are used for classification, and it is evident that the CNN-LSTM algorithm exhibits superior accuracy compared to alternative deep learning algorithms. The model is provided with video clips of a certain period, and it distinguishes the clip by analyzing the sequence of motions exhibited by the driver in the video. The key objective of this work is to promote road safety by notifying drivers when they exhibit signs of drowsiness, minimizing the probability of accidents caused by fatigue-related disorders. It would help in developing an ADAS that is capable of detecting and addressing driver tiredness proactively. The work intends to limit the potential dangers associated with drowsy driving, hence promoting enhanced road safety and a decrease in accidents caused by fatigue-related variables. The work aims to achieve high efficacy while maintaining a non-intrusive nature. This work endeavors to offer a non-intrusive solution that may be seamlessly included into current automobiles, hence enhancing accessibility to a broader spectrum of drivers, through the utilization of facial movement analysis employing CNN-LSTM and U-Net-based architecture.

Keywords: Artificial Intelligence; Advanced Driver Assistant Systems; Internet of Things; U-Net; Automated Vehicles; Convolutional Neural Networks_Long Short-Term Memory

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Journal Not Specified* **2023**, *1*, 0.

Received:
Revised:
Accepted:
Published:

1. Introduction

In recent years, the rapid advancement of technology has led to the convergence of various fields, giving rise to innovative solutions that enhance safety, efficiency, and convenience in everyday life. One such intersection is the realm of IoT and automotive safety, where the application of IoT principles to automobiles has paved the way for groundbreaking advancements in driver assistance systems [1]. Among these innovations, the detection of driver drowsiness holds immense significance in preventing accidents

and ensuring road safety. The IoT-assisted automatic driver drowsiness detection system offers a multi-tiered approach to road safety. Firstly, it operates in real-time, providing instantaneous feedback to the driver and triggering alerts if drowsiness is detected. These alerts can take various forms, such as auditory alarms, haptic feedback, or even in-vehicle adjustments to lighting and climate control. Secondly, the system contributes to data-driven insights by collecting and analyzing a wealth of information over time. This data can be utilized for statistical analysis, identifying trends, and refining algorithms to enhance detection accuracy.

A vehicle is the most powerful thing on the road. When used recklessly, it may be dangerous and occasionally the lives of other road users may be at risk. Failure to realize when we are too tired to drive is one type of carelessness. Many academics have authored study papers on driver tiredness detecting systems to monitor and prevent a disastrous outcome from such recklessness. As a result, this work was carried out to provide a new viewpoint on the current situation and optimize the solution. The most significant cause of accidents at night is driving when tired. Fatigue and drowsiness are regularly to blame for serious accidents on roadways [2] [3]. Only identifying the tiredness and warning the driver will solve this issue. Drivers are more prone to fall asleep on trips requiring long stretches of driving on regular routes, such as highways. High-risk trips are the ones made for work-related purposes, particularly those involving truck drivers. Another high-risk travel option [4][5] is corporate car drivers. As a result, there is a clear relationship between the time of day and the likelihood of falling asleep behind the wheel. These observations highlight the interplay between driving circumstances, specific driver profiles, and the temporal aspect of drowsy driving risk. Recognizing these correlations is essential for developing effective strategies to mitigate the dangers associated with driver drowsiness and enhance overall road safety.

The global public health issue of motor vehicle collisions (MVC) and injuries is global [6]. Drivers' sleepiness and weariness are major contributing factors to fatal crashes and MVC risk factors. The research is well-referenced about the prevalence of driver fatigue, drowsiness, and weariness, as well as its effects on the incidence of MVC and injuries from traffic accidents. The pattern of acute weariness, exhaustion, persistent sleepiness, sleep issues, and high workload has been connected to poor performance in psychomotor tests and driving simulators due to the rising incidence of MVC, injuries, and deaths in specific populations. A well-liked instrument for gauging self-reported driving behavior and finding a connection between it and accident involvement is the Driver Behaviour Questionnaire (DBQ) [7]. As human mistakes cause most traffic accidents, DBQ is one of the most often used research instruments to explain erroneous driving behaviors in three basic categories, including errors, infractions, and lapses. They then suggested incorporating the study's findings into micro-simulations to more precisely imitate drivers' actions on urban street networks. Drowsy driving impacts everyone regardless of age, profession, economic situation, or level of driving expertise. Drivers frequently feel tired, and there are occasions when people have to drive while being severely sleep-deprived. Teenagers and new drivers have spent less time on the road. Thus their driving skills have not yet matured. Younger drivers are also more inclined to drive after hours for social or professional reasons, which increases their risk of driving while fatigued. Shift and night workers frequently put in long hours at the office; they're usually worn out when it is time to clock out. They don't require a long journey home, yet many still try to get to their cars out of habit and duty. The risk of sleepy driving is six times higher for people who work night, rotating, or double shifts than for other categories of workers. Doctors, nurses, pilots, police officers, and firefighters are just a few occupations that frequently need long shifts. Compared to the typical commuter, people who drive for a living log more kilometers on the road. Because many commercial drivers work long hours and face strict deadlines, they also have a considerably higher risk of driving while fatigued. Regular business travelers are especially vulnerable to the dangers of drowsy driving because they frequently experience jet lag and switch time zones as frequently as they do ZIP codes. Getting enough sleep cannot be easy if people travel

a lot for work to stay safe on the road. For drivers with sleep disorders, drowsy driving can be a daily struggle. Some drivers may experience daytime exhaustion and drowsiness due to narcolepsy or insomnia, but those with untreated obstructive sleep apnea (OSA) are significantly more at risk of doing so. Some drugs can also have the opposite effect, causing sleepiness in drivers when they need to be focused behind the wheel.
Sleep-related crashes are more likely to result in catastrophic injuries, possibly due to the higher speeds involved and the driver's incapacity to avoid an accident or even stop in time [8]. Drowsiness can be understood in many ways, like the tendency to yawn, sleepiness, tiredness, and others. This causes a significant number of fatal accidents and deaths. It is currently a hot topic for research.

In summary, this paper seeks to advance road safety by detecting driver drowsiness and issuing timely alerts, thereby reducing the risk of accidents linked to fatigue. The research leverages IoT and deep learning technologies to create a system that is not only effective but also unobtrusive, making use of facial movement analysis to improve driver safety on the road. Ultimately, the goal is to make this solution easily accessible to a wider range of drivers, thereby contributing to safer roadways and a reduction in accidents caused by drowsy driving. Section 1 presents an introduction of the paper. Section 2 presents the contribution of the paper. Section 3 presents the related work on various drowsiness detection using different machine learning and deep learning techniques. Section 4 designs the architecture and mathematical analysis of the proposed model. Section 5 describes the result analysis and discussion. The final section summarizes our research findings and future planning.

2. Contribution of the Paper

By addressing a critical safety concern on the road, this study will help reduce accidents caused by drowsy driving. This study can substantially impact road safety by combining IoT, deep learning, and facial movement analysis to detect driver drowsiness automatically. The Contributions of the paper are given below:

- The paper presents U-Net-based segmentation to only take information from the physical regions of the driver's body. After Segmentation, we encode the image information and combine data from multiple time steps. Then minimize the effect of an external factor, and U-Net-based Segmentation is carried out before passing the frames of the model.
- After that, the segmented body region is fed to the CNN-LSTM model, which gives a softmax output indicating the driver's probability of being drowsy.
- The method uses a combination of segmentation, image feature extraction, and time series analysis algorithms to make the classification decision confidently.
- The paper leverages the IoT principles to develop a real-time monitoring system. By strategically placing sensors within the vehicle and utilizing interconnected data transmission, the paper pioneers a practical application of IoT technology in the context of driver safety.
- The paper identifies and addresses challenges associated with accurate drowsiness detection, such as minimizing false positives and negatives and accommodating various driving scenarios.

3. Related Works

Driving while tired increases the likelihood of a collision or accident. Many individuals are killed in automobile accidents yearly due to sleepy driving caused by a lack of sleep, drunkenness, drug and alcohol abuse, heat exposure, or drinking. Accurate drowsiness detection based on eye state has been achieved using a variety of indicators and parameters as well as the expertise of specialists. An essential component for sleepiness detection is predicting facial landmarks, detecting eye states, and presenting the driver status on the screen. Major traffic accidents frequently occur when the driver feels tired from long hours of driving, a physical sickness, or even alcoholism. Drowsiness can be defined as a natural

state where an individual feels exhausted. The individual's reflex is significantly reduced, which can cause the driver to become unable to take quick actions when necessary. Also, studies have shown that driving performance also worsens with increased drowsiness. A human can quickly tell if someone is tired by detecting specific actions or behaviors. Drowsy driving is a serious issue that affects the driver, puts other people's lives in danger, and harms the nation's infrastructure. There has been an enormous surge in the daily use of private transportation in this modern society. When traveling a distance for an extended period, driving will become dull. Traveling for a long time without getting any rest or sleep is one of the key reasons drivers lose focus.

The detection method follows eye movement and facial expression to identify the drowsiness state of the driver with the help of Convolutional Neural Networks (CNN). If people look into recent times, a method that helps in behavioral recognition by understanding the upper body postures and giving that state of the image as output is also being processed by Convolutional Neural Networks (CNN). In that proposed recognition model, some data were collected related to driving. Another proposed model detected whether the person was busy with phone calls and one hand was holding the steering wheel. It is based on the Faster-RCNN mechanism. Another model is based on an attention mechanism entirely different from the CNN modes of the tool. This attention mechanism proposed model uses the classification of fine-grained images. However, these mechanisms do not help predict the driver's drowsiness while driving and do not focus on the distracting scenes inside the vehicle. Therefore, it is difficult to identify the driver's actions while going. Another proposed model identifies the position of faces through various poses. Existing methods for detecting driver drowsiness can be categorized into three kinds: Physiological, vehicle-based, and behavioral.

The first method attaches a device to the driver's skin. Awais et. al in [10] exploit the use of ECG and EEG characteristics. First, it collects EEG features such as time domain statistical descriptors, complexity metrics, and power spectrum measures. ECG also eliminates heart rate, HRV, LF/HF ratio, and other variables. Next, all of these features are combined using SVM, and discrimination is achieved by utilizing these hybrid features.

Another method by Warwick et. al [12] used the idea of bio-harness. The system works in two phases. Fig: 1 represents the various drowsiness detection techniques. The driver's physiological data is gathered in the first phase using bio-harness. An algorithm analyses the readings in the second phase. The problem with the methods of this category is that some device has to be attached to the drivers' skin, which may only be comfortable for some. The second category analyses the usage pattern of the vehicle control system, like steering wheel movements, braking habits, and lane departure measurements. Here also, different methods use these data to detect driver drowsiness.

Zhenha et. al [13] suggested steering wheel motions over time using a temporal detection window as the primary detection feature. This window is used to evaluate the steering wheel's angular velocity during the time-series analysis by comparing it to the statistical properties of the movement pattern below the fatigue threshold.

Li et. al [14] used the Steering Wheel Angles (SWA) data to monitor driver drowsiness under natural conditions. The problem with these vehicle-based methods is that they need to be more reliable. As a result, it can significantly affect the road's nature and the driver's driving skills. This may result in many false positives. The third category is more reliable than the second one, as it only focuses on the driver.

The method by Saradadev et. al [15] used the mouth and yawning as the detection feature. First, it locates and tracks the mouth using a cascade of classifiers, and then an SVM model is used to analyze and classify a drowsy driver.

Another method by Teyeb et. al [17] analyses the eye closure and the head posture for discrimination. First, the face is partitioned into three regions. Then the Wavelet Network is used to determine the state of the eyes.

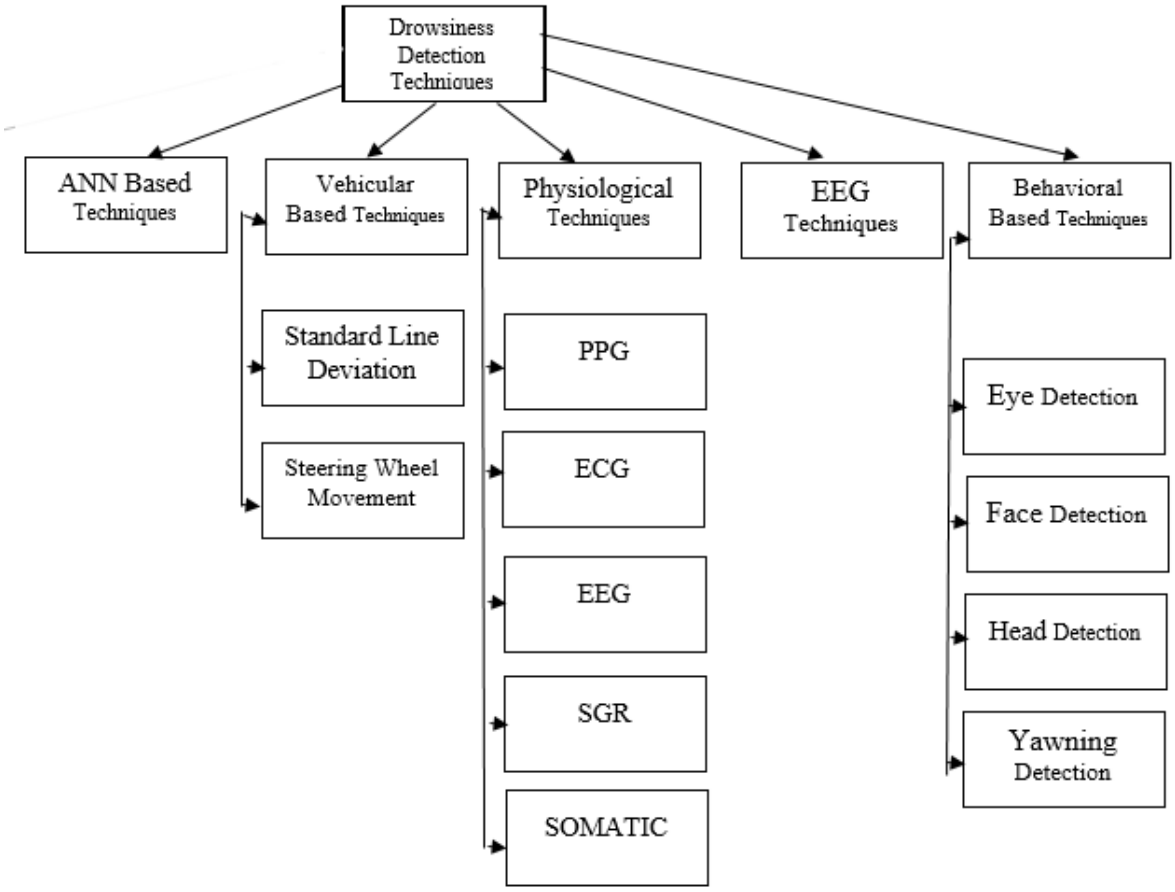

Figure 1. Various drowsiness detection techniques.

The authors in [18] propose a video-based driver sleepiness detection using real-time techniques with 91.7 % accuracy and represented the Karolinska sleepiness scale. They have compared their model and PERCLOS-based baseline method detection.
Alam et. al proposed in [19] a deep learning technique based on a convolution neural network for drowsy driver detection using signal channel EEG signal.
In [20], the authors proposed an EEG classification system for Driver Drowsiness based on Deep Learning techniques with 90.42% accuracy. They have designed two procedures such as data acquisition and model analysis. Slow eye closure is often a reliable method for detecting drowsiness, which can be captured by measuring PERCLOS, i.e. the percentage of eye closure. Issues like different lighting conditions or orientations may influence the system. Upon the occurrence of this, the system is blind. The automatic detection of driver fatigue using EEG signals and deep neural networks is a multidisciplinary effort, combining expertise in neuroscience, signal processing, and machine learning. It has the potential to significantly contribute to road safety by preventing accidents caused by drowsy driving.
Sobhan et. al [22] presented a mechanism designed to identify driver fatigue, a critical factor in mitigating traffic accidents. The approach involved collecting information from 11 individuals, resulting in a comprehensive dataset adhering to established standards. The study's findings indicate that the proposed deep CNN-LSTM network demonstrated the ability to hierarchically learn features from the raw EEG data, surpassing the accuracy rates achieved by previous comparison methods in the two-stage classification of driver fatigue.
Dua et. al proposed in [21] different deep learning models like Alexnet, VGG-Facenet, FlowimageNet, and ResNet using SoftMax classifier with 85% accuracy. Jamshidi et. al. in [25] proposed hierarchical deep drowsiness detection to detect driver drowsiness with 87.19% and used the LSTM network for temporal information between the frames. The

authors proposed a hybrid learning technique, namely NTHU-DDD, UTA-RLDD.
Liu et. al [26] used Deep Neural networks as essential in such a model as in the Machine Learning field. It has high demand and great value. Hussein et. al [23] presented a study that uses three deep-learning-based algorithms, Deep Neural Network, Recurrent Neural Network, and CNN, to categorize captured driving data using the standard identifying procedure, and choose the best one for a proposed detection mechanism. Several approaches were employed to avoid overfitting. The CNN outperformed the other two classification algorithms, with an accuracy of 96.1%, and was thus suggested for the recognition system. In [28], the algorithm works in two stages. First, it locates and crops the mouth region. Then in the second stage, an SVM is used to classify if the image presents driver fatigue and alerts the driver accordingly. The system uses a cascade of classifiers to locate the mouth region. The SVM is trained on pictures of mouth regions representing various styles of yawning, but to start with, our method takes much more information as input, not just the mouth region. Based on the research study, the motive is to control the rate of accident cases due to fatigue or driver drowsiness so that no mishaps occur with the person and, most importantly, to enhance the safety in traffic rules and regulations whenever reckless driving takes place due to the unconscious state of mind, i.e., drowsiness. The neural network is trained using the PERCLOS and POM drowsiness thresholds [29]. MT-CNN extracts the face and the feature points, which aids in obtaining the shape of the eyes and mouth. Next, EM-CNN takes action by assessing the condition of the eyes and mouth. When a threshold is met or surpassed, eye and mouth closure degrees are determined by observing the unbroken picture frames; these segmented images of the driver are passed through blocks of convolutional layers followed by a 1x1 Conv. Coated for dimension reduction, the output is passed through an LSTM layer. Zhang et. al [53] proposed AdaBoost, LBF, and PERCLOS algorithms, and the accuracy of the model is 95.18%. The hardware and software required for this method are relatively inexpensive, making it a feasible solution for mass deployment in the paper. Ulrich, L et al. studied [31] 11 participants participated in an auditory and haptic ADAS experiment while having their attention tracked while they were driving. The drivers' faces have been captured using an RGB-D camera. These pictures were then examined through the use of a deep learning technique, which involves training a convolutional neural network (CNN) expressly to recognize facial expressions (FER). Studies have been conducted to evaluate potential connections between these findings and ADAS activations as well as event occurrences, or accidents. Different algorithms for driver drowsiness detection are given below in Table 1. Table 2 presents the research gap of the existing algorithms on drowsiness detection.

Table 1. Different algorithm on driver drowsiness detection

Paper	Algorithms	Accuracy	Advantages	Disadvantages
Li et. al [37]	SVM (Support Vector Machine)	91.92%	Can be integrated with other driver assistance systems	Requires training to interpret EEG data and May be affected by other factors, such as stress or fatigue.
Pauly et. al [38]	Histogram of oriented gradient and Support Vector Machine	91.6%	This method can detect drowsiness in real-time, so it can provide early warning signs to the driver.	The SVM classifier needs to be trained on a dataset of images of drowsy and non-drowsy drivers in order to be effective.
Flores et. al [45]	Viola-Jones object detection Adaboost algorithm, Neural Networks and Support Vector Machine	-	This system only requires a camera to detect drowsiness, so it is non-intrusive to the driver.	This system may be affected by other factors, such as driver distraction or fatigue.
B.Manu et. al [46]	Viola Jones algorithm K-means algorithm SVM	94.58%	This method is accurate in detecting drowsiness, even in challenging conditions.	This method may be affected by other factors, such as driver distraction or fatigue and environmental factors.
Rahman et. al [47]	Viola Jones algorithm AdaboostHaar classifier	94%	Eye blink monitoring has the potential to reduce the number of accidents caused by driver drowsiness.	Eye blink monitoring may be affected by other factors, such as driver distraction or fatigue.
Anjali et. al [48]	Viola-Jones object detection Haar cascaded classifier	-	This strategy has the potential to minimize the number of accidents caused by tiredness in the driver.	The system needs to be trained on a dataset of eye blink data from drowsy and non-drowsy drivers in order to be effective.
Coetzer et. al [49]	Artificial neural networks, Support Vector Machines, Adaptive boosting (AdaBoost)	98.1%	Challenging conditions such as low lighting and different head poses.	Eye detection may be affected by environmental factors such as lighting and occlusion.
Punitha et. al [51]	Viola-Jones Face Cascade of Classifiers Support Vector Machine	93.5%	Eye state analysis has been shown to be accurate in detecting drowsiness	Ambient elements such as illumination and occlusion, may have an impact on eye state analyses.

Table 2. Research gap of the different algorithm on driver drowsiness detection.

Paper	Approach	Key Contribution	Research Gap
Mungra et. al (2020) [66]	CNN-based Emotion Recognition	High accuracy in detecting fear, anger, and sadness expressions.	Limited investigation on the impact of different CNN architectures and data augmentation techniques.
Weng et. al (2022) [68]	Multimodal Emotion Recognition	Improved accuracy through the multimodal fusion of facial expressions and signals.	Lack of focus on temporal analysis and integration of deep learning architectures like LSTM.
Lea et. al (2017) [70]	Temporal Convolutional Networks	Real-time emotion recognition with accurate fear, anger, and sadness detection.	Limited exploration of combining CNN and LSTM for improved emotion detection.
Li et. al (2019) [72]	LSTM-based Facial Expression Recognition	Consideration of temporal context for improved emotion detection.	Absence of spatial analysis and utilization of U-Net architecture for accurate facial feature extraction.
Li et. al (2020) [73]	Attention Mechanism and CNN	Enhanced discriminative power through an attention mechanism.	Insufficient exploration of combining attention mechanisms with LSTM.
Anand et. al (2019) [74]	U-Net Architecture for Facial Analysis	Precise facial feature extraction and localization.	Limited investigation on temporal dynamics and utilization of LSTM for improved emotion detection.
Wang et. al (2015) [75]	Facial Expression Recognition in Vehicles	Robust emotion detection addressing challenges of occlusions and partial views.	Lack of exploration of multimodal fusion and comprehensive temporal analysis for improved accuracy.

The suggested method's segmentation algorithm, [33] U-Net, is simply a collection of convolution and ReLU blocks with some max pool layers in between the first and second halves, followed by some transpose convolution layers. U-Net is characterized by its U-shaped architecture, which consists of a contracting path (encoder) followed by an expanding path (decoder). This unique design enables it to capture both high-level context and fine-grained details in an image. Drowsy face detection often requires analyzing facial features at multiple scales, as signs of drowsiness can manifest differently in different parts of the face. U-Net's encoder-decoder structure and skip connections enable the network to extract features at various levels of granularity, allowing it to recognize drowsy faces with diverse characteristics. U-Net's ability to handle inputs of varying sizes and adapt to different lighting conditions, poses, and backgrounds makes it robust to real-world image variability. Drowsy face detection systems often need to work in diverse environments, and U-Net's flexibility can help maintain performance across these settings. U-Net typically converges quickly during training, which is beneficial for training drowsy face detection models. Rapid training can save time and computational resources, making it easier to experiment with different model architectures and training data variations.

U-Net's efficiency in terms of both training and inference makes it suitable for real-time applications, such as drowsy driver detection systems. This ensures timely warnings or interventions when drowsiness is detected. The ability of U-Net to capture subtle facial cues and context can help reduce false positives in drowsy face detection. This ensures that alarms are triggered only when genuine signs of drowsiness are present, enhancing the user experience and avoiding unnecessary interruptions. U-Net is a powerful and versatile architecture for drowsy face detection in image-to-image mapping tasks. Its ability to capture spatial information, extract multi-scale features, and adapt to varying conditions contributes to the accuracy and reliability of drowsy face detection systems, making them valuable for driver safety and other applications where monitoring facial expressions is critical. Finally, in terms of operation, the transposed convolution multiplies the filter value by the encoded matrix to produce another padded matrix with a more excellent resolution. This stage displays the output of the LSTM layer, which may be combined with another system to create a functional end-to-end system. For example, if a buzzer is linked at the end and the driver is identified as tired, the buzzer will sound, or in a self-driving car, the car will safely stop on the side of the road and then do something to wake the user up. D Gao et. al [62] described Federated learning based on Connection Temporal Classification (CTC) for the Heterogeneous IoT. Federated learning involves training machine learning models across decentralized devices while keeping data on the devices, addressing privacy and communication challenges. In this paper, we propose FLCTC, a federated learning system based on CTC for heterogeneous IoT applications. We built FLCTC and a particular solution for forest fire prediction to illustrate its applicability. This integration enhances the capabilities of both IoT and ML, enabling intelligent decision-making, automation, and insights from the vast amount of data generated by IoT devices.

Temporal analysis is crucial in various applications, and integrating deep learning architectures like LSTM (Long Short-Term Memory) can indeed enhance the ability to model temporal dependencies in data. LSTMs are particularly effective in handling sequences and time-series data due to their ability to capture long-range dependencies. The combination of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks is a powerful approach, especially for tasks like emotion detection. CNNs are excellent at extracting spatial features from data, while LSTMs excel at capturing temporal dependencies. In the context of emotion detection, a common approach is to use CNNs to extract relevant features from input data (such as images or sequences of frames), and then feed these features into an LSTM for capturing temporal dynamics. Combining attention mechanisms with LSTM is a fantastic avenue for improving the performance of models dealing with sequential data. Attention mechanisms enable the model to focus on specific parts of the input sequence, making it more adaptable and effective in capturing relevant information. The "DistB-SDCloud" architecture, which improves cloud security for intelligent IIoT applications, is presented in the study [39]. In order to maintain flexibility and scalability while offering security, secrecy, privacy, and integrity, the suggested architecture employs a distributed BC technique. Clients in the industrial sector profit from BC's efficient, decentralised, and distributed environment. The paper also presented an SDN technique to enhance cloud infrastructure's resilience, stability, and load balancing.

The paper [40] proposes a lightweight and robust authentication system for WMSN, which is made up of physically unclonable functions (PUFs) and state-of-the-art blockchain technology, to address these two major concerns. Furthermore, a fuzzy extractor approach is presented to handle biometric data. Two security evaluation techniques are then employed to demonstrate the great reliability of our suggested approach. Lastly, among the compared systems, the suggested mutual authentication protocol has the lowest computing and communication cost, as demonstrated by performance evaluation trials. Zhou at al. [41] presented the domains of two input vehicle images are transformed into other domains in the network structure using a generative adversarial network (GAN)-based domain transformer. A four-branch Siamese network is then created to learn two distance metrics between the images in the two domains, respectively. In order to calculate the

ultimate similarity between the two input photos for vehicle Re-ID, the two distances are finally merged. The outcomes of the experiments indicate that the suggested GAN-Siamese network architecture attains cutting-edge results on four extensive vehicle datasets: VehicleID, VERI-Wild, VERI-Wild 2.0, and VeRi776. Zhou, Z et al. [43] identify the boundary frames as possible accident frames based on the generated frame clusters. Next, in order to verify whether these frames are indeed accident frames, the paper record and encode the spatial relationships of the items identified from these potentially accident frames. The comprehensive tests show that the suggested method satisfies the real-time detection requirement in the VANET environment and provides promising detection efficiency and accuracy for traffic accident detection. Zhou, Z et al. [44] introduced a novel identity-based authentication system. The paper proposed method demonstrates secure communication between various components of the green transport system through the use of lightweight authentication mechanisms. Zhou, Z et al. [41] for HAR, the paper suggests a robust subspace clustering (SOAC-RSC) scheme that is based on sequential order-aware coding. Two expressive coding matrices are learned in a sequential order-aware manner from unconstrained and restricted films, respectively, by feeding the motion properties of video frames into multi-layer neural networks to generate the appropriate affinity graphs. Khajehali et. al [69] presented a complete systematic literature review focusing on client selection difficulties in the context of federated learning. The goal of this SLR is to support future CS research and development in FL. Deng et. al [64] presented an iterative optimization approach on EE under the condition of interference constraint and minimal feasible rate of secondary users. To begin, the Dinkelbach method-based fractional programming is used with a given UAV trajectory to determine the appropriate gearbox power factors. In the second step, the successive convex optimization technique is used to update the system parameters using the prior power allocation scheme. Finally, to find the optimal UAV trajectory, reinforcement-learning-based optimization is used. Sarkar et. al [67] the Industrial Internet of Things (IIoT) that has gained importance at a time when the medical industry's potential is rapidly rising (see also [76], [77]). To solve this difficulty, this paper presents the Intelligent Software-defined Fog Architecture (i-Health). Based on each patient's past data patterns, the controller decides whether to transport data to the fog layer.

The fusion of IoT technology and facial movement analysis has led to the creation of an innovative solution for enhancing driver safety. By harnessing real-time data acquisition and advanced machine learning techniques, the IoT-assisted automatic driver drowsiness detection system has the potential to significantly reduce accidents caused by driver fatigue. As technology continues to evolve, this system represents a testament to the power of interdisciplinary collaboration in creating impact solutions that shape the future of road safety. Here, our contribution extends to the novel data acquisition methodology by capturing a range of facial movements and expressions, including eye closure duration, blinking patterns, and head orientation. In this way, we are able to acquire real-time data crucial for accurate drowsiness detection.

4. Proposed Model

Drowsiness is a big issue while driving; therefore, some drowsiness detection must be implemented in front of a driver while they are driving a vehicle. So, with the help of libraries named OpenCV and Dlib, we develop a driver drowsiness detection system that will initiate whether the person's eyes are closed or opened in the software model, i.e., the eyes are in an active state or passive (lazy) state. Moreover, the main motive is identifying or detecting if the person is yawning at the steering wheel. It becomes essential to implement such a detection system to reduce the accidents that occur due to fatigue caused by being tired or sleepy. This fatigue becomes more dangerous at night when the accident cases increase at a rate of more than 50 percent. So, to reduce the number of these road accidents, this advanced method of detection system must come to active implementation in the real-life world. Based on the research study, the motive is to control the rate of accident cases due to fatigue or driver drowsiness so that no mishaps occur with

the person and, most notably, to enhance the safety in traffic rules and regulations whenever the reckless driving takes place due to unconscious state of mind, i.e., drowsiness. The detection method follows the physical nature, i.e., eye movement and facial expression, to identify the drowsiness state of the driver with the help of Convolutional Neural Networks (CNN). The proposed model uses a 15-second video clip as input. The video is sampled at the 1-second interval, which yields 15 frames. These frames are then passed through a U-Net to extract the region of interest (ROI), in this case, the driver's body. Using 1x1 Conv layers significantly reduces the dimension of the output we get from the convolution layers, which plays a significant role in encoding the features extracted from the input frames.

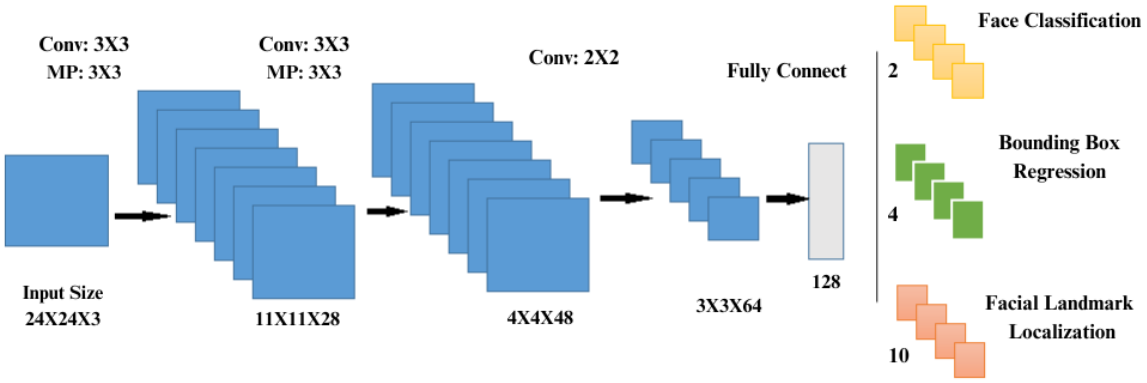

Figure 2. P-Net architecture.

U-Net has been a very successful model when it comes to image-to-image mapping. Detecting face is a very intensive task that can be challenging in Real-world situations as there are variations in driver posture and dissipated environment aspects such as radiance or occlusions. Using the Depth-Cascading Multitasking Framework, we can easily align and detect Faces which improves the internal relation by using facial features like position, right and left eye, the mouth's corners, and the nose. By the architecture of Multitask Cascaded Convolutional Networks, we can understand that it compares P (Proposal), R (Refined), and O (Output)-Net. These three sub-networks detect Face and Feature points. In P-Net, Different-sized image pyramids are assembled in a chronology as input. Whether a face includes 12x12 at each position is determined by the Convolutional Network. Then a boundary box is calibrated with a regression vector to remove overlapping face regions. Fig. 2 represents the architecture of P-Net, while in Fig. 3 the R-Net image reshaped to 24x24 is shown. Boundary box regression and non-maximum value suppression shield the face window. A connection layer is added to the network structure to acquire an accurate face position. O-Net image is reshaped to 48x48 to prevail the final face position along with face features. To unify all real-world pictures of different sizes, the convolution layer is used, which resizes them into 175x175, and Pools also acquires a 44x44x56 sized feature map.

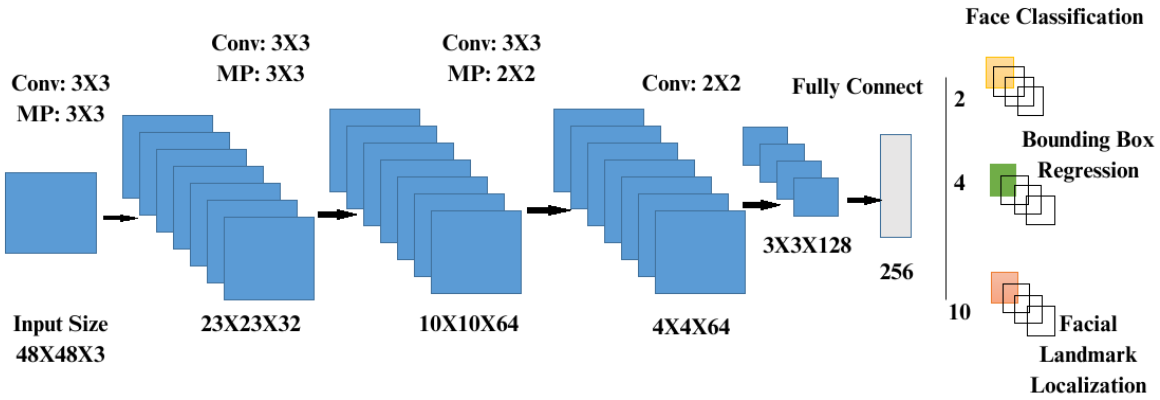

Figure 3. R-Net architecture.

The Convolution layer of the Convolution kernel is 3x3 along with 1 step size, while the Pooled layer is 3x3 and the step size is 2. A pixel layer is used to eliminate size reduction, which causes a loss of details in the borders. Now three pooling layers boot up the adaptability by 3x3 pooling layers like 1x1, 3x3, and 5x5. Another pooling map is the 44x44x256 feature map. Then an 11x11x72 feature map is generated by channeling through three layers of the residual block. A one-dimensional vector by feature map and linked layer is used to reduce the parameters by random inactivation to minimize overfitting. Using softmax, we can now define eyes and mouth as opened or closed. Although there is a similar network for time series data GRU, Long Short Term Memory (LSTM) is better at retaining information longer, which helps associate specific patterns in embedding the frames from the video clip. Lastly, the final block is composed by fully connected layers followed by softmax activation. Fig. 4 represents our proposed U-Net-based Architecture. So the convolution kernel's convolution layer is 3x3 with 1 step size, whereas the pooled layer is 3x3 with two steps. Before an operation, a pixel layer is employed in the edges to avoid size reduction, which causes a loss of detail in the borders. Three pooling layers, such as 1x1, 3x3, and 5x5, now boost flexibility by 3x3. The 44x44x256 feature map is another pooling map. Finally, by channeling through three levels of the residual block, an 11x11x72 feature map is formed. A one-dimensional vector with a feature map and connected layer is utilized to lower the parameters by random inactivation to prevent overfitting. Random inactivation, also known as dropout, is a regularization technique commonly used to prevent overfitting in neural networks. We can specify whether the eyes and lips are open or closed with softmax. Although a comparable network exists for time series data GRU, Long Short Term Memory (LSTM) is superior at keeping knowledge for extended periods, which aids in associating specific patterns in embedding the frames from the video clip. Finally, the last block consists of ultimately linked layers followed by softmax activation. Fig. 4 depicts our suggested U-Net architecture.

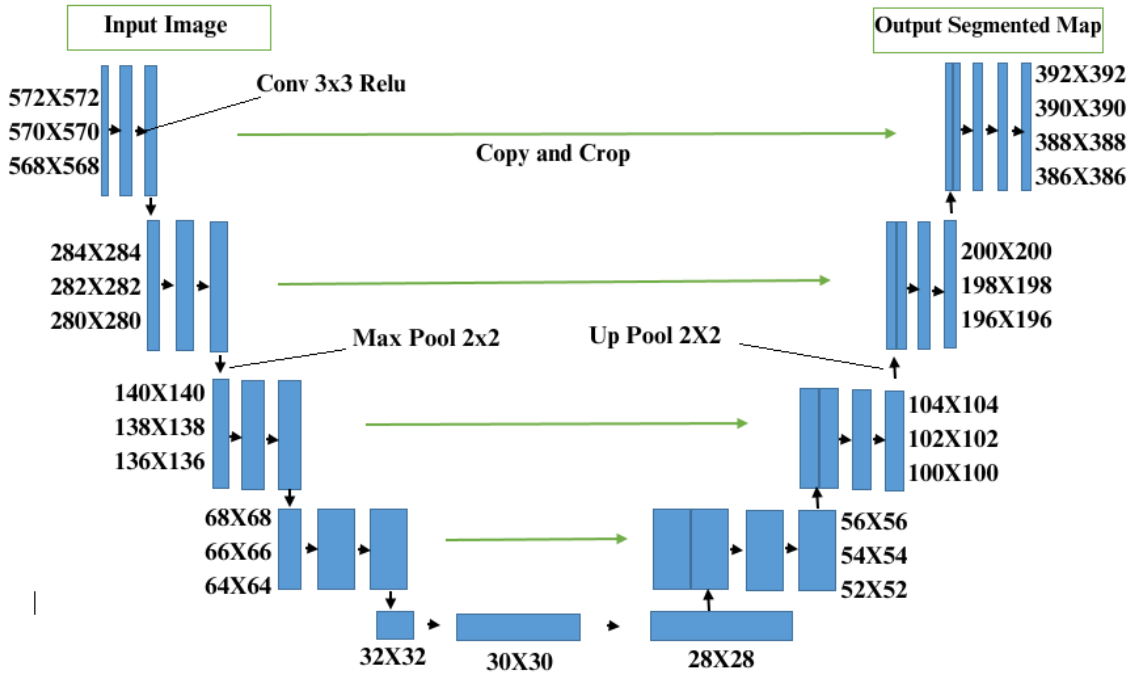

Figure 4. Proposed U-Net based architecture.

The frames from the images are mapped to a segmentation map, as shown in Fig.4, where the driver's body is mapped as one entity, and everything else visible in the image is marked as another entity. Then the segmentation map is used as a mask to extract the driver's body from each frame. The U-Net's name is justified by its architecture's shape, which also resembles that of an autoencoder.

Figure 5. Application of 1x1 filters.

The first half of U-Net captures the context with a compact feature map. The second symmetric half is there for precise localization to retain the spatial information to compensate for the down-sampling and max-pooling done in the first stage. The convolution layers in our model Conv. Nets embeds the information from the frames to connect the last LSTM layers to make a meaningful classification. The output from the Conv. traps sometimes gets vast in the channel dimension, which creates the requirement for relatively more computation in the later stages. This can be solved by using 1x1 Conv. filters. As shown in Fig. 5, the 32x32x512 output can be combined with a 1x1x512 filter to reduce the output dimensions to 33x32x1. The reduced output from the 1x1 convolution layer is fed into an LSTM layer.

The overall effectiveness of the segmentation result contributes directly to the accuracy and reliability of the entire driver drowsiness detection system. Regular testing and evaluation on diverse datasets and under various conditions are essential to ensure that the segmentation process meets the desired performance standards.

4.1. Dataset and Data Pre-Processing

We collected the database database that I. Nasri et al developed from Kaggle. The dataset size was around 5.3 GB, and the total dataset time was approximately 7.3 hours. The system gets frontal pictures of the driver, which correspond to the previous 55 seconds taken at 12 frames per second by a camera mounted on the car. The videos featured a frame rate ranging from 12 to 27 frames per second with a resolution of 640x480. We can build up the experimental environment with more than 3144 photos. We intended to train and test in four different scenarios. We trained over 489 video images for Open Eyes and tested over 243 video images. 384 for closed looks, 266 for testing, and 384 for testing. Now that the mouth is open, there are 496 photos for testing and 320 images. The feature data in this paper vary significantly in scale. In this work, the data set was normalized to remove the impact of the hierarchy between the features. The normalizing procedure aids in increasing computational accuracy, preventing gradient explosion during network training, and hastening the loss function's convergence.

4.2. Feature Extraction

Both CNN and LSTM are well-known deep learning models. The CNN network can extract the data features in the spatial dimension by layer-by-layer learning the local features of the data using the hidden layer. The LSTM network can extract features in the temporal dimension and has the qualities of long-term retention of contextual historical knowledge. The activation function, optimization function, and learning rate of the CNN-LSTM neural network module are all set to Relu, Rmsprop, and 0.0001 respectively in the CNN-LSTM network. The samples are first entered into the CNN in the CNN-LSTM model, and then two sampling operations and four one-dimensional convolution operations are carried out. The input sample's tensor is (None, 41, 64), where None stands in for the input sample's size. The obtained (None, 10, 128) tensor is fed into the LSTM network following the CNN procedure. The amount of output features is adjusted by altering the number of nodes in the first dense layer, which serves as the middle layer for extracting features. Finally, the last fully connected dense layer produced the (None, (5-40)) tensor. By modifying the epoch value, learning rate, and number of nodes in the dense layer during model training, the ideal model parameters are discovered.

4.3. Detection of Drowsiness State

To monitoring PERCLOS can be part of a system to detect when a driver may be becoming drowsy, which is crucial for preventing accidents on the road. Drowsiness detection is a complex task, and combining multiple indicators often leads to more accurate results. PERCLOS, when used in conjunction with other measures, contributes to a more comprehensive and reliable drowsiness detection system. The Human Body System reflects its states automatically. EM-CNN uses these kinds of human physiological reactions to evaluate PERCLOS and POM [16]. The equation of PERCLOS with percentage is given below in Eq. (1) [24].

$$PERCLOS = (\sum_{i}^{N} f_i / N_f) \times 100\% \quad (1)$$

$\sum_i^N f_i$ Constitutes frames of closed eye per unit time. N_f is the total frames per unit and f acts for the frame of the closed eye. To calculate the threshold of drowsiness, a collection of 13 video frames is used to test and evaluate the value of Perceptron Learning Rule with Output Scaling (PERCLOS). In Eq. (2) when its value gets 0.25 or greater that means the eye is in its closed state for a continuous time which expresses the drowsiness. The neural network is trained based on the drowsiness threshold of PERCLOS and POM [14]. Recurrent Input Output (RIO) refers to a neural network architecture or a specific type of layer that utilizes recurrent connections. PERCLOS, in the context of neural networks,

is not a commonly used acronym. However, it could potentially be interpreted as a term related to a neural network architecture or algorithm that combines the perceptron learning rule with output scaling. The perceptron learning rule is a fundamental concept in neural networks, and output scaling may refer to adjusting the output of neural network layers to match a desired range or format. POM, in the context of neural networks, could be interpreted as a "Probabilistic Output Model." This might refer to a type of neural network or model designed to provide probabilistic predictions or estimates as outputs. For example, probabilistic neural networks or certain types of Bayesian neural networks can produce probabilistic outputs, which are valuable in tasks like uncertainty estimation or probabilistic classification. MT-CNN extracts the face along with the feature points, which helps get the RIO of the eyes and mouth. Then EM-CNN comes into action by evaluating the state of the eyes and mouth. Observing the uninterrupted image frames, eye and mouth closure degrees are calculated when a threshold is matched or exceeded. The segmentation algorithm used in the proposed method, [15] U-Net, is essentially just a bunch of convolution and ReLU blocks with some max pool layers in-between in the first and second half convolution and ReLU blocks followed by some transpose convolution layers. The two halves are also connected with multiple skip connections between them. Convolutional, ReLU, and max-pooling layers are also used in the primary model, specifically in the CNN part of the CNN-LSTM architecture [16]. The equation can describe the working of the convolution operation in Eq. (2) [11].

$$\sum\sum I(ip+p, j+r).K_r(p,r) \tag{2}$$

Here I indicates the input matrix and K indicates the 2D kernel of size p×r. In Eq. (3), the convolution blocks also use ReLU activation function to add non-linearity to the output. Working of ReLU can be described as: Where f is a function of x.

$$f(x) = maximum(0, x) \tag{3}$$

Max pool is the most commonly used method among all the pooling layers and all it does is reducing the number of parameters by sampling the maximum activation value from a patch of the image or the matrix. The working of max pooling can be described as given in Eq. (4),

$$P_{i,j} = maximum(f(x) : x = A_{i+m,j+n}) \tag{4}$$

where A is the activation output from ReLU and P is the output from the max pool layer. The U-Net also used the transposed convolution operation, which is similar to max pooling but it up-samples the encoding instead. In Eq. (5) transposed convolution takes in an image of size i×i and a kernel of size k×k and outputs an up-sampled matrix of dimension:

$$(i-1) \times s - (2 \times p) + (k-1) + 1 \tag{5}$$

where s is the stride of the padding. As far as the operation of the transposed convolution goes, it just multiplies the value of the filter with the encoded matrix to get another padded and higher-resolution matrix. This stage presents the output from the LSTM layer that can be used with another system to make an end-to-end helpful system. Maybe a buzzer is

connected at the end, which goes on when the driver is detected to be drowsy, or in the case of a self-driving car, the car may safely park on the side of the road and then do something to wake the user up. Fig. 6 presents the architecture of the CNN-LSTM model.

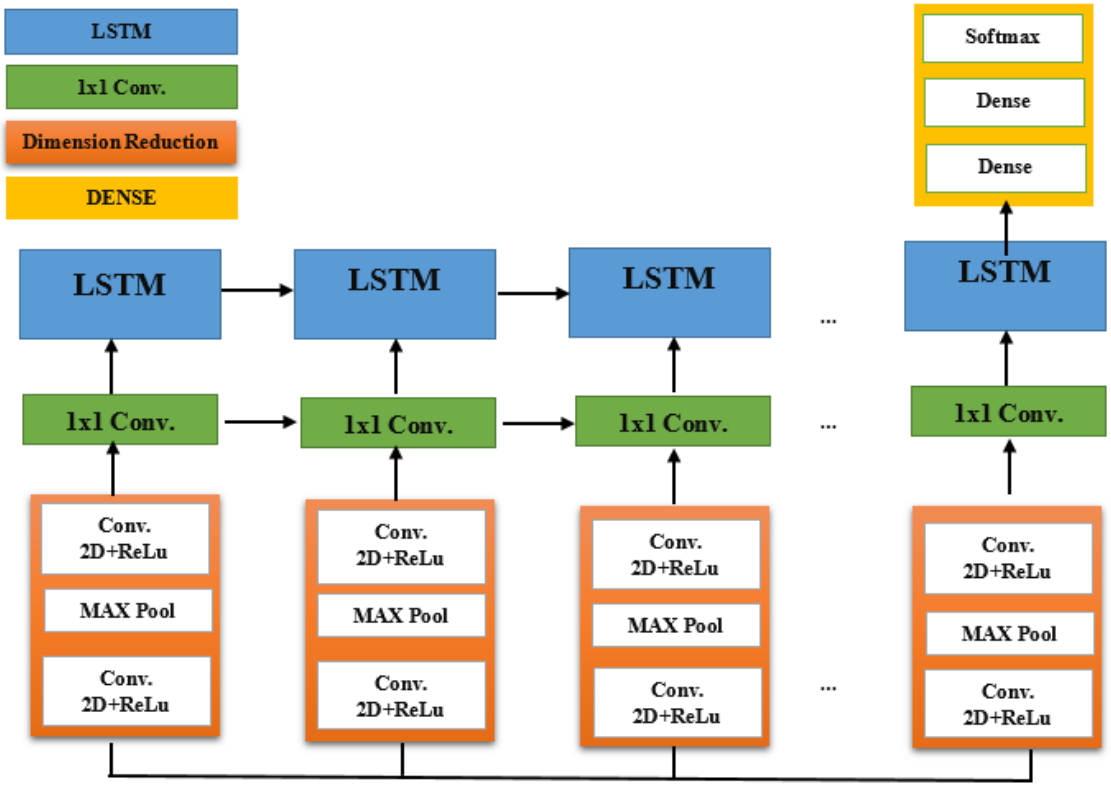

Figure 6. CNN-LSTM architecture.

The CNN-LSTM model uses LSTM layers to fuse past time steps. When retaining data activation effects for much longer in the recursion, LSTM layers are proven to be much more effective than GRU layers. The main reason for such a difference is that LSTM uses three gates to update the memory cell. One is the update gate, also present in GRU, and the other is the forget and output gates. More formally, the three gates can be described by the following equation [32]:

$$\tau_{update} = \sigma(W_u[a^{t-1}, x^t] + b_u \tag{6}$$

In the above equation and in the subsequent equations a^{t-1} denotes the activation from the previous time step, x^t denotes the input in the current time step. W_u and b_u are the parameter matrix and the bias respectively and τ_{update} is the value of the update gate. Then:

$$\tau_{forget} = \sigma(W_f[a^{t-1}, x^t] + b_f \tag{7}$$

where W_f and b_f are the parameter matrix and the bias respectively and $_{forget}$ is the value of the update gate.

$$\tau_{output} = \sigma(W_o[a^{t-1}, x^t] + b_o \tag{8}$$

In (8), W_o and b_o are the parameter matrix and the bias respectively and $_{output}$ is the value of the update gate.

The memory cell of the LSTM is calculated using the following equation:

$$c^{<t>} = \tau_{update} * c1^{<t>} + \tau_{forget} * c^{<t-1>} \tag{9}$$

where $c^{<t>}$ and $c^{<t-1>}$ is the value of the memory cell. $c1^t$ is the candidate for the memory cell that is supposed to replace the current one? Here * means vector multiplication. The value for $c1^t$ can be written in terms of this equation:

$$c1^{<t>} = tanh(W_c[a^{t-1}, x^t] + b_c \tag{10}$$

In Eq. (11), finally, the current activation is calculated by combining the output gate and $c^{<t>}$. Here * means vector multiplication.

$$a^{<t>} = \tau_{update} * c^{<t>} \tag{11}$$

Detecting drowsiness in drivers is a challenging task that demands a sophisticated approach. One promising solution is the fusion of CNN and LSTM networks, two powerful deep-learning architectures. This fusion capitalizes on the strengths of both CNNs, known for their image analysis prowess, and LSTMs, renowned for modeling sequential patterns. The CNN-LSTM architecture holds the potential to revolutionize drowsiness detection systems, making roads safer and saving lives. By analyzing real-time video streams of a driver's face, this hybrid model can not only capture intricate facial features but also track temporal patterns in driver behavior. This innovative approach has the capability to accurately determine when a driver is becoming drowsy, thus enabling timely interventions and preventive measures.

In this system, CNNs are employed as the first line of defense, extracting meaningful features from images of the driver's face. These features are then handed over to the LSTM network, which specializes in understanding the sequence of these features over time. By learning from historical patterns, the LSTM can distinguish between normal behavior and signs of drowsiness, such as drooping eyelids, yawning, or erratic facial movements. The strength of this combined architecture lies in its ability to consider not only the current frame but also the context provided by preceding frames. This context-awareness capability allows the model to identify subtle changes that might escape a single-frame analysis. As a result, the CNN-LSTM model can adapt to the dynamic nature of drowsiness, which often manifests gradually rather than abruptly. Through rigorous training on diverse datasets encompassing various lighting conditions, driver characteristics, and scenarios, the CNN-LSTM model refines its ability to accurately recognize drowsiness. The model's high accuracy, sensitivity, and specificity make it an indispensable tool in modern driver assistance systems. Its potential applications extend beyond just drowsiness detection – it can be integrated into smart vehicles, fleet management systems, and transportation infrastructure, contributing to a safer and more secure transportation ecosystem. The steps of the proposed algorithm are here below reported:

Step: 1 Preprocessing of Image (M) datasets.
Step: 2 Images combined with input from trained models.
Step: 3 Retrieve the result of the final convolution layer of the model that was provided.
Step: 4 Flatten the n dimensions, decreasing their number to n-1.
Step: 5 Apply different layers of CNN-LSTM

Padding (Conv2d): The formula below should be used to determine the padding width, where pd stands for padding and fd for filter dimension: $fd \in Odd$,

$$pd = \frac{fd - 1}{2} \tag{12}$$

Forward propagation: It is separated into two phases. After computing the intermediate value K that is produced by convolution of the input data from the preceding layer with M tensor it then adds bias b and applies nonlinear activation function on intermediates value:

$$K^l = M^l.AF^l + b^i, AF^l = g^l(k^l) \tag{13}$$

Max-pooling: The output matrix's proportions can be calculated using (14) while accounting for padding and stride:

$$n_{output} = \frac{n_{output} + 2pd - ft}{s} + 1 \tag{14}$$

The cost function's partial derivative is shown as :

$$\partial AF^l = \frac{\partial l}{\partial AF^l}, \partial K^l = \frac{\partial l}{\partial K^l}, \partial M^l = \frac{\partial l}{\partial M^l}, \partial b^l = \frac{\partial l}{\partial b^l} \tag{15}$$

After applying the chain rule in (15):

$$\partial K^l = \partial AF^l \times g(K^l) \tag{16}$$

Sigmoid activation function, linear transformation and Leaky ReLU are as follows:

$$f(r) = \frac{1}{1 + e^- r}, K = M^t.R + b, f(r) = (0.01 \times r, r) \tag{17}$$

It returns r if the input is positive and 0.01 times r if the input is negative. As a result, it also produces an output for negative values. This minor modification causes the gradient on the graph's left side to become nonzero.

Apply Softmax function: The neural network typically does not create even one final figure. To represent the likelihood of each class, these numbers must be reduced to integers from zero to one.

$$\sigma(m)_j = \frac{e^m j}{\sum_{p-1}^{p} e^m j} for j = 1..p \tag{18}$$

Applying CNN-LSTM: LSTM has been used after CNN has been applied, i.e. CNN-LSTM:

$$input_t = \sigma(w_i[h_{t-1}, r_t] + b_i)$$

$$\text{forgetgate}_t = \sigma(w_{function}[h_{t-1}, r_t] + b_{forgetgate}$$

$$\text{output}_t = \sigma(w_{output}[h_{t-1}, r_t] + b_{output}) \tag{19}$$

where x_t input at current timestamp is h_{t-1} is the previous LSTM Block. σ represents the sigmoid function. $forgetgate_t$ is forget gate. b is bias for respective gates. Where c_t is the cell state at timestamp (t). $\widetilde{c_t}$ represents a candidate for cell state at timestamp (t):

$$\widetilde{c_t} = tanh\ tanh(w_c[h_{t-1}, r_t] + b_c,$$
$$\text{c}_t = f_t * c_{t-1} + i_t * \widetilde{c_t}, h_t = o_t * tanh\ tanh(c_t) \tag{20}$$

In the context of driver drowsiness, this hybrid CNN-LSTM architecture presents a formidable tool. By analyzing real-time video feeds of a driver's face, the CNN component is capable of discerning crucial facial features, such as eye movement, blink rate, and facial expressions. These features, extracted through the convolutional layers of the CNN, serve as a rich foundation of visual cues. Fig. 7 presents the architecture of the proposed model. However, what sets this architecture apart is its LSTM component. This network structure possesses the ability to understand sequences and patterns within the extracted features. In the realm of drowsiness detection, this implies that the system can not only assess the current state of the driver's face but also track how that state evolves over time. Subtle cues that might signify drowsiness, such as prolonged eye closure or micro expressions, can be identified by the LSTM as part of a sequence, enabling more accurate and reliable detection. The predictive power of the CNN-LSTM architecture extends beyond mere real-time analysis. By leveraging the LSTM's memory capabilities, the model can recognize trends and tendencies that indicate an increasing likelihood of drowsiness. This forward-looking approach allows for timely intervention, such as alerts to the driver or automated adjustments to vehicle settings, preventing potential accidents before they occur.

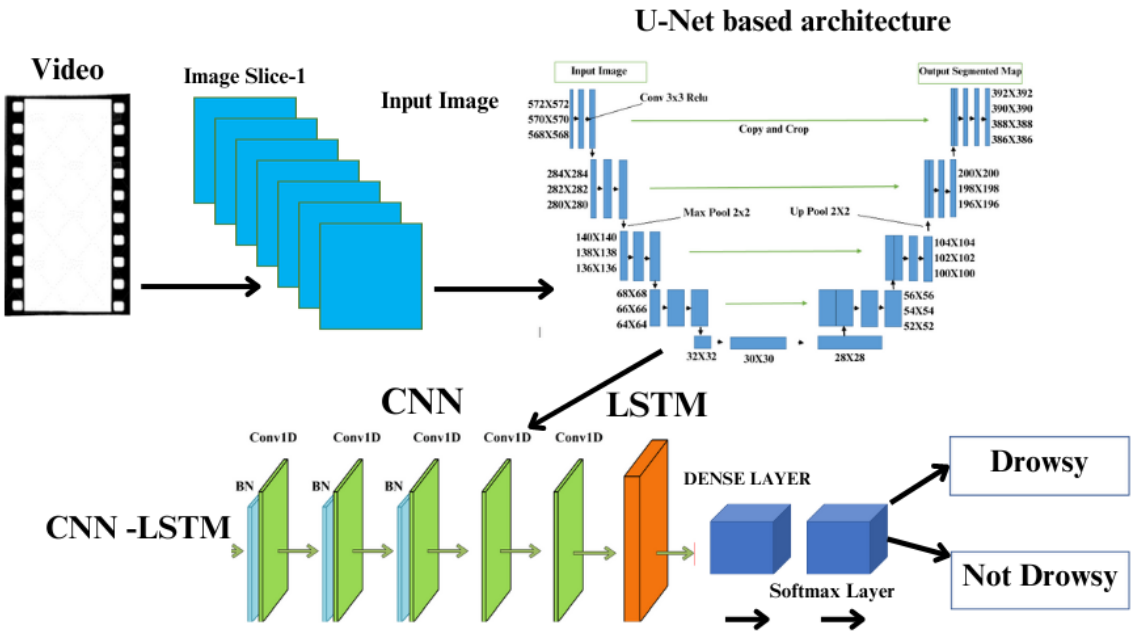

Figure 7. Proposed model architecture.

CNN-LSTM network is a popular architecture used in deep learning for tasks that involve both spatial and temporal data, such as video analysis or sequential image data. Input data for CNN-LSTM typically consists of a sequence of images or tensors, where each element in the sequence represents a frame in a video or a timestamp in a time series. The input data is first passed through a CNN to extract spatial features. The CNN layers consist of convolutional layers followed by pooling layers, which help capture important spatial patterns in the data. These layers are responsible for feature extraction from individual frames. After the CNN layers, it can either flatten the features into a 1D vector or use global average pooling to reduce the spatial dimensions. This step depends on the specific problem and the architecture used. The output of the CNN is then fed into an LSTM network, which is responsible for capturing temporal dependencies and patterns across the sequence of frames. The LSTM network consists of multiple LSTM layers. Each LSTM cell maintains an internal state that can capture information from previous time steps. This internal state helps in modeling long-term dependencies in the data. The LSTM layers process the sequence of features generated by the CNN, one-time step at a time, and update their internal states accordingly. The final LSTM layer is often connected to one or more fully connected (dense) layers, which can be used for making predictions or classification. The output layer's architecture depends on the specific task. For example, for video classification, it might consist of a softmax layer for class probabilities. The MSE (Mean Squared Error) function is utilized as the loss function. To update the settings of each network layer, the common Adam optimization technique is employed as the optimizer. The Dropout layer that was used contributed to the model's enhanced generalization, decreased training time, and prevention of overfitting. In this research, the constructed model's prediction performance was also compared with that of EM-CNN, VGG-16. GoogleNet, AlexNet, and ResNet50 models in order to confirm the model's efficacy. These methods were chosen for comparisons because of their following characteristics: (i) EM-CNN is a semi-supervised learning algorithm which uses only weakly annotated data and performs very efficicntly for face detecttion; (ii) VGG-16 is a 16-layer deep neural network. A relatively extensive network with a total of 138 million parameters, that can achieve a test accuracy percentage of 92.7 in ImageNet, a dataset containing more than 14 million training images across 1000 object classes; (iii) GoogleNet

is a type of CNN based on the Inception architecture. It utilises Inception modules, which allow the network to choose between multiple convolutional filter sizes in each block; (iv) AlexNet uses an 8-layer CNN and showed, for the first time, that the features obtained by learning can transcend manually-designed features, breaking the previous paradigm in computer vision; (v) ResNet-50 is a 50-layer CNN (48 convolutional layers, one MaxPool layer, and one average pool layer), that forms networks by stacking residual blocks.

5. Experimental Results

We aim to identify drowsiness, awaken users to prevent accidents, and produce an alarm sound and app notification. The proposed approach produced results that were greater than 98% in terms of accuracy. For this project, we needed real-world images of drivers while driving. This real-world environment helps to build the architecture, which makes an accurate model. Fig. 8 represents training and validation accuracy of the training Dataset.

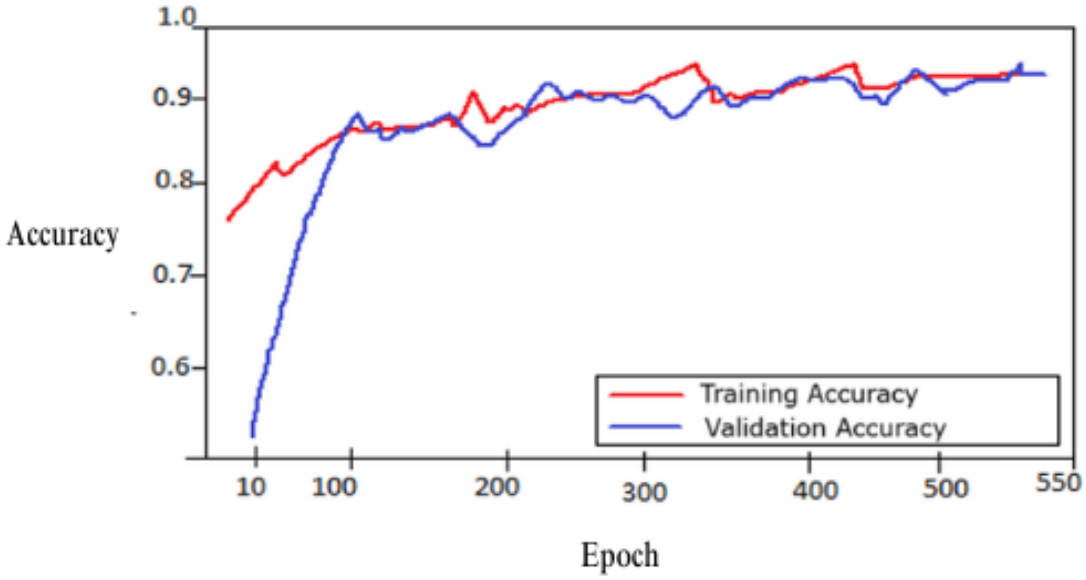

Figure 8. Training and validation accuracy.

Last but not diminutive, the case of mouth in the closed state was represented by 640 images for testing and 475 for testing. Some sample images on routine and drowsiness detection during day and night light are shown in Fig. 9 and Fig. 10.

Figure 9. Sample normal and drowsiness images (daylight).

Figure 10. Sample normal and drowsiness images (night light).

5.1. Implementation Process and Results Discussion

Emotion-CNN (EM-CNN) represents a cutting-edge approach to emotion recognition, leveraging the power of Convolutional Neural Networks (CNNs) to decode and understand the nuanced expressions of human emotions within visual data. In a world where artificial intelligence continues to bridge the gap between human experiences and computational capabilities, EM-CNN emerges as a specialized architecture designed to decipher the complex language of emotions embedded in images and videos. Emotion-CNN (EM-CNN) represents a cutting-edge approach to emotion recognition, leveraging the power of Convolutional Neural Networks (CNNs) to decode and understand the nuanced expressions of human emotions within visual data. In a world where artificial intelligence continues to bridge the gap between human experiences and computational capabilities, EM-CNN emerges as a specialized architecture designed to decipher the complex language of emotions embedded in images and videos. Visualizing the evolution of Convolutional Neural Networks (CNNs), the VGG-16 architecture emerges as a pivotal milestone in the realm of image classification and computer vision. Conceived by the Visual Geometry

Group at the University of Oxford, VGG-16, standing for Visual Geometry Group with 16 layers, represents a breakthrough in the pursuit of deep learning excellence. VGG-16 has 16 weight layers, including 13 convolutional layers and 3 fully connected layers. It is known for its simplicity with small 3x3 convolutional filters and deep architecture, which aids in feature learning.

In the dynamic landscape of deep learning, GoogleNet, or Inception v1, represents a groundbreaking convolutional neural network architecture developed by researchers at Google. Introduced in 2014, it was designed to address challenges associated with computational efficiency, parameter reduction, and the ability to capture diverse features across multiple scales. GoogleNet is deeper than VGG-16 but uses a more efficient architecture with 1x1 convolutions for dimension reduction and parallel filter operations. This reduces the number of parameters and computational cost. AlexNet is a deep convolutional neural network (CNN) featuring eight layers, five convolutional layers followed by three fully connected layers. The architecture's depth was a departure from previous shallower networks, allowing it to learn intricate hierarchical features from raw image data. AlexNet consists of five convolutional layers followed by max-pooling layers and three fully connected layers. It helped establish the effectiveness of deep learning in computer vision tasks.

ResNet50, a variant of the ResNet (Residual Network) architecture, stands as a monumental advancement in deep neural networks, particularly in addressing the challenges associated with training very deep networks. The defining feature of ResNet50 lies in its introduction of residual learning. In traditional deep networks, the optimization process can be hindered by the vanishing gradient problem, making it challenging for information to flow through the network. Residual learning addresses this by introducing shortcut connections, or skip connections, allowing the network to learn residual functions. ResNet50's skip connections mitigate the vanishing gradient problem, allowing the training of very deep networks. This architecture is widely used in image classification and other computer vision tasks. CNN-LSTM, a hybrid neural network architecture, represents a sophisticated integration of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. This fusion is designed to harness the strengths of both convolutional and recurrent architectures, making it particularly well-suited for tasks that involve both spatial and temporal dependencies. In many real-world applications, data is not only spatially rich but also exhibits temporal dependencies. For instance, in video analysis, understanding the content requires capturing both spatial features (such as objects and patterns within frames) and temporal dynamics. CNN-LSTM emerges as a solution to address this dual challenge. The CNN part is used for spatial feature extraction, capturing patterns in different spatial regions, while the LSTM part handles the temporal dependencies by processing sequences of features. One thing that is easy to identify is that when a person is driving, and he is on average amplitude, then the changing of mouth state is less. But on the other side, it is harder to define an eye blinking state as machine vision calculates the closure state. To simplify this process, a detailed flowchart is shared to help to clarify the idea. Using Erosion and Expansion binaries, RIO is smoothened, which helps to define the state of the eye. Black pixels allow for representing the binocular image area. The count of black pixels is also essential. The threshold value is set to 0.15, which defines the eye closure state. A value exceeding 0.15 is considered an eye in an open forum. 32 batch size is proposed for our training. Selecting the number of epochs and the batch size for drowsy detection using a neural network involves considering various factors related to the dataset, model architecture, and computational resources. Drowsy detection often involves intricate patterns and temporal dependencies. The number of epochs should be sufficient to allow the model to learn and capture these complex patterns in the data. Experiment with different epoch values and observe the model's behavior on both training and validation sets. Drowsy detection often involves analyzing sequences of data over time, such as video frames or time-series data from sensors. Smaller batch sizes might allow the model to capture temporal dependencies more effectively. If the drowsy detection system is intended for real-time applications, consider a batch size that balances model

accuracy and inference speed. Smaller batch sizes can lead to faster predictions, which is crucial in real-time scenarios. The selection of epochs and batch size for drowsy detection involves a balance between capturing temporal dependencies, computational efficiency, and preventing overfitting. Experimentation and close monitoring of model performance are essential for making informed decisions. It takes special consideration to implement a neural network on hardware. Both training and testing could be done with more RAM and a powerful processor. During our training, we trained our Model with 100 epochs. The number of epochs and the batch size were chosen empirically, according to Fig. 8, as the best trade-off between the accuracy level and the computational complexity required by the investigated algorithm. At the same time, we are talking about the system configuration of a computer with 6 Core Processors, 16GB RAM, and Nvidia GTX 1650Ti GPU on 64-bit Windows 10. This processing power was adequate for the operation of our application. The suggested approach uses Google's "Colab Pro Plus version" as the execution platform. Conversely, MT-CNN, EM-CNN, and CNN-LSTM are done on Python 3.10 and Keras 2.4.0 with Tensorflow 2.70 as the environment.
We can see that the training accuracy of EM-CNN, VGG-16, GoogleNet, AlexNet, ResNet50, and our proposed Model is the percentage of 86.54, 92.46, 66.19, 46.12, 56.09, and 98.70 and testing accuracy of EM-CNN, VGG-16, GoogleNet, AlexNet, ResNet50, and our proposed model is the percentage of 89.54, 92.4, 66.19, 48.50, 56.09 and 98.80. Table 3 defines the training accuracy and testing accuracy of the different networks.

Table 3. Experimental training and testing results.

Network	Training Accuracy(%)	Testing Accuracy(%)
EM-CNN	86.54	86.54
VGG-16	92.46	92.46
GoogLeNet	66.19	66.19
AlexNet	46.12	48.5
ResNet50	56.09	56.09
Proposed Model (CNN-LSTM)	**98.7**	**98.8**

Table 4. describes the precision, recall, F1-score, and accuracy of GoogLeNet, ResNet50, AlexNet and VGG-16, and EM-CNN and CNN-LSTM. CNN-LSTM presents better results than other different learning algorithms. TensorFlow is then used to translate the 10.5 hours of video data in the dataset into frames. The research employed various measures to evaluate how well deep learning models could detect driver sleepiness. These metrics encompassed accuracy, loss, precision, recall, and F1-score. To elucidate the model performance, a confusion matrix, depicted in Fig. 11, was frequently provided. This matrix serves as a table to gauge the accuracy of a deep learning model across different dataset types using a test dataset. In the pursuit of training robust and accurate models for drowsy detection, the optimization of loss functions stands as a crucial compass guiding the neural network towards learning the intricate patterns indicative of drowsiness. The selection of an appropriate loss function is akin to fine-tuning the model's compass, aligning it with the landscape of the drowsy detection task. Here, we explore the essence of this journey, understanding the key considerations and pathways in loss function optimization. In the grand expedition of drowsy detection, the optimization of loss functions becomes a compass of precision, guiding the neural network through the diverse and challenging terrains of imbalanced data, temporal dependencies, and nuanced pattern recognition. With each epoch, the model refines its navigation skills, inching closer to the destination of heightened accuracy and vigilance in drowsy detection. The loss function defines the objective that the model aims to minimize during training. The experiment's loss function is categorical cross-entropy, which is provided by:

$$loss = -\sum_{i=1}^{N} x_{a,b} Inp_a, b \tag{21}$$

where N is the number of classes; x is a binary indication indicating whether or not c is the accurate prediction for observation a; and p is the anticipated probability that the observation belongs to class b. Each deep learning model completes roughly 35 epochs throughout the fit process, with a batch size of 32.

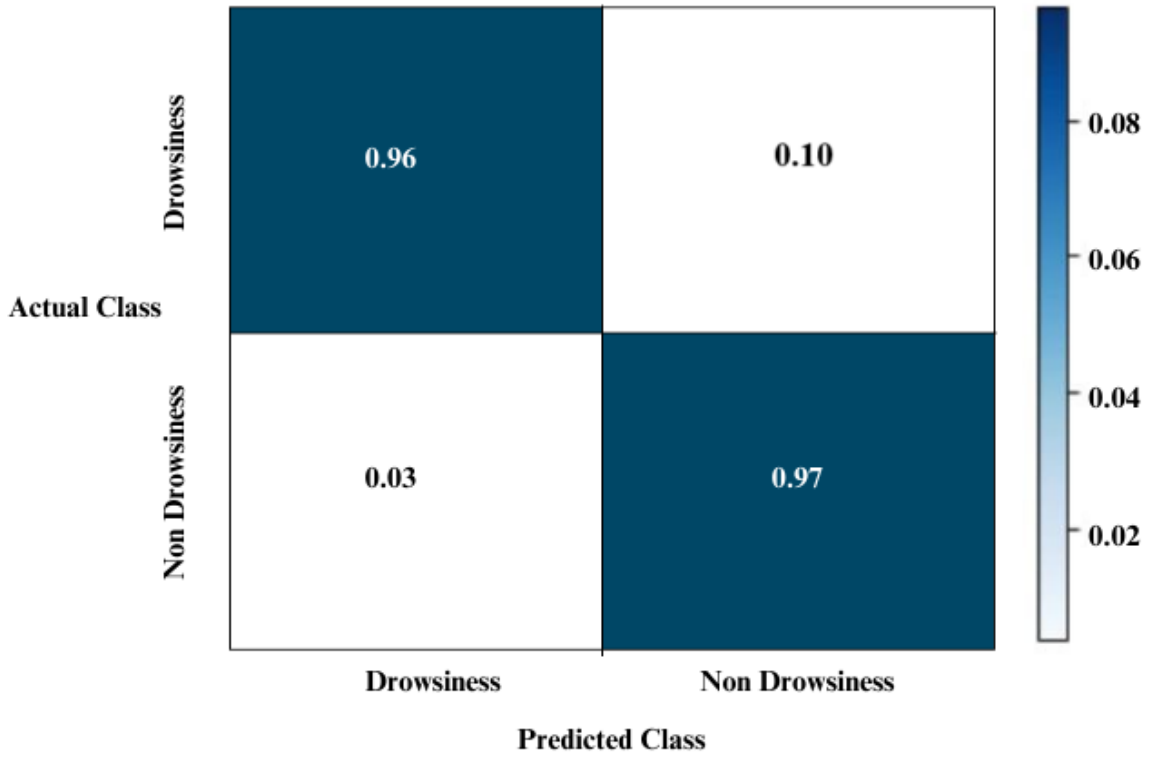

Figure 11. Confusion matrix.

The confusion matrix for driver drowsiness detection using a CNN-LSTM model would be a table that shows the performance of the model in terms of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). It helps evaluate the accuracy of the model in classifying instances of driver drowsiness.
True Positive (TP) is the model that correctly predicts a drowsy driver. True Negative (TN) is the model that correctly predicts a driver as not drowsy. False Positive (FP) is the model that incorrectly predicts a driver as drowsy when they are not. False Negative (FN) is the model that incorrectly predicts a non-drowsy driver as drowsy. In Equation 13, Precision is the measure of the accuracy of positive predictions. It is the ratio of true positives to the total instances predicted as positive.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{13}$$

In Equation 14, Recall is the measure of the ability of the model to capture all relevant instances. It is the ratio of true positives to the total actual positive instances.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

(14)

In Equation 15, the F1 Score is the harmonic mean of precision and recall. It provides a balanced measure that considers both false positives and false negatives, particularly useful when there is an imbalance between classes.

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

(15)

In Equation 16, accuracy is the measure of the overall correctness of a model. It is the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

(16)

Accuracy is the percentage of samples correctly categorized by the classifier to all samples for a given test dataset, or the test dataset's accuracy when the loss function is 0-1. The loss function is used to gauge how well a model predicts, and the lower it is, the better. The class in question is typically seen as the positive class, whereas other classes are seen as the negative class. Sensitivity is the ability of a classification model to correctly identify true positive instances from all actual positive instances. Specificity is the ability of a classification model to correctly identify true negative instances from all actual negative instances.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \text{Recall}$$

(26)

$$\text{Specificity (\%)} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

(27)

When we compare one Model with another, we can appreciate its efficiency. We have compared our CNN-LSTM with other methods like GoogLeNet, ResNet50, AlexNet VGG-16, and EM-CNN. After comparing the entire procedure, the EM-CNN model proves its efficiency by outperforming with 97.46% accuracy, a sensitivity of 97.67%, and a specificity percentage of 78.21%. We can see the specificity of all techniques in Fig. 12.

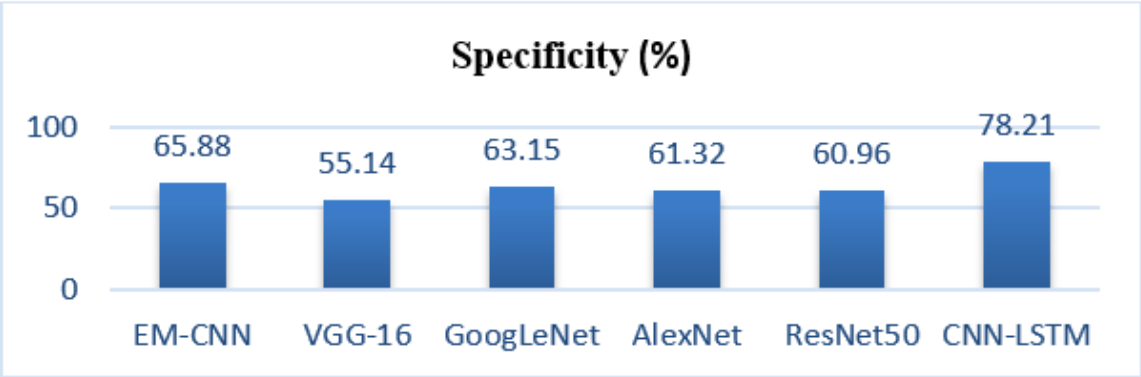

Figure 12. Specificity(%) of different deep learning model

Table 4. Results of evaluation deep models using Precision, Recall, F1-score and Accuracy

Network	Precision	Recall	F1-Score	Accuracy(%)
EM-CNN [57]	0.721	0.685	0.602	96.62
VGG-16 [58]	0.113	0.218	0.117	82.98
GoogLeNet [59]	0.708	0.611	0.594	94.01
AlexNet [60]	0.461	0.485	0.312	91.48
ResNet50 [61]	0.677	0.536	0.514	93.85
Proposed Model (CNN-LSTM)	**0.819**	**0.652**	**0.732**	**98.46**

Figure 13 depicts the Loss value of a single scenario after numerous training steps (epoch) with learning rates of $1x10^{-}4$ and $1x10^{-}5$ respectively. The loss value is shown on the y-axis, and epochs are represented on the x-axis. The accuracy steadily improves as the time lengthens and the Loss value drops. We finish training and continue to the testing phase when the epoch approaches 35 and We got the loss value of different deep learning models in Table 4.

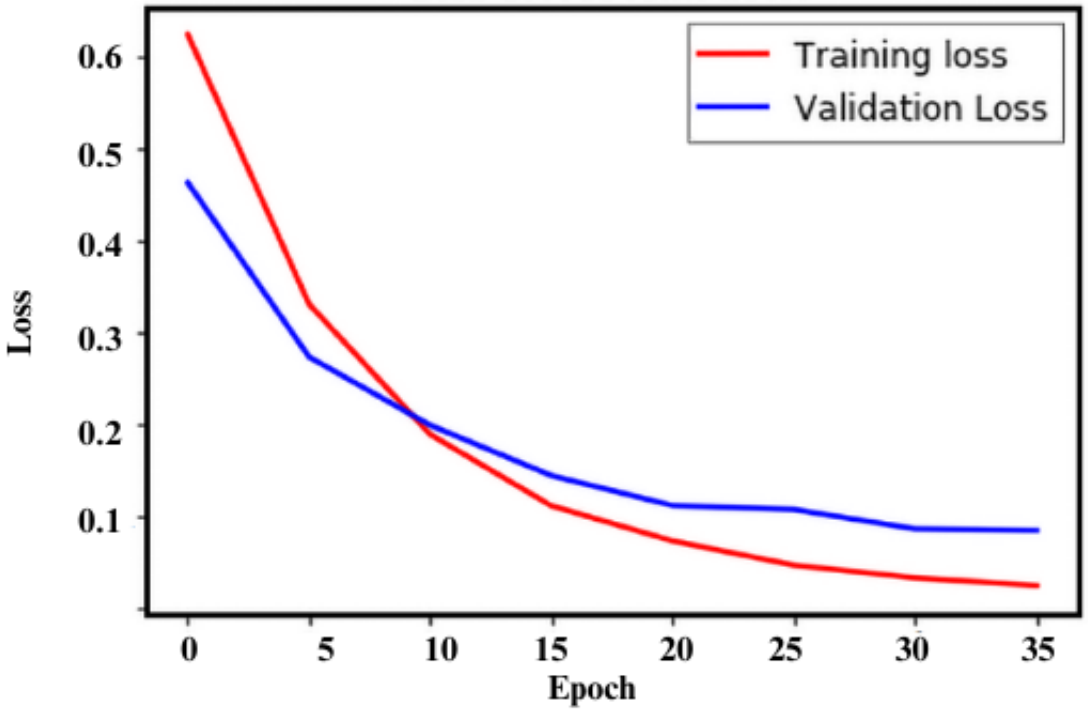

Figure 13. Loss value.

After thoroughly testing and comparing, we can conclude that CNN-LSTM is more accurate and sensitive to the state of mouth over the state of eyes. Also, the mouth displays more precise drowsiness, which is a good sign. In [36], the following AUC for classifications of CNN-LSTM. Using a temporal correction system, we can detect eye blink frequency, which helps to identify the drowsiness from a video frame. For an eye, the opening state is this gets 1; when it is closed, that turns into 0, so for the blinking frequency, it will be a sequence of 1 and 0. Now it is time to pull the trigger of the threshold, as we can't use the results of current results.

Table 5. Accuracy (%) of no of dataset with AUC (%)

Dataset(Images)	Accuracy	Eye Close	Eye Open	Mouth Close	Mouth Open
5	98.32	99.22	99.11	99.65	99.34
15	97.34	95.72	94.28	96.95	99.56
35	98.43	97.32	99.21	99.35	99.34
65	98.12	97.37	94.29	96.95	98.94
95	98.33	98.32	99.29	99.45	99.24
115	98.44	94.89	96.91	97.69	92.38
135	98.34	98.32	99.21	99.45	99.34
165	98.22	98.61	96.67	98.91	98.76
195	98.22	97.39	94.73	96.74	96.14
235	97.11	98.32	99.24	99.45	99.24
265	97.18	94.89	96.56	97.69	92.38
285	97.22	98.32	99.67	99.45	99.34
345	97.26	98.62	99.21	99.68	99.14
365	98.43	98.32	99.21	99.45	99.34
385	98.12	98.61	96.67	98.91	98.76
400	98.33	97.39	94.73	96.74	96.14
415	97.35	97.31	94.35	96.47	98.34
425	98.11	98.32	99.21	99.45	99.34
445	97.35	98.32	99.20	99.45	99.34
455	98.43	98.34	99.27	99.45	99.35
465	98.12	98.34	99.21	99.56	99.39

For this project threshold value of PERCLOS as POM must be extracted. For frame-by-frame recognition, we extracted images from 15 video frame sequences. Accuracy of Different states of AUC such as Eye open, Eye closed, mouth closed, and mouth open for classification of CNN-LSTM presented in Table 5. Table 5 describes the accuracy of CNN-LSTM and AUC of different datasets. Models from AlexNet [60] are utilized as comparison standards. In model size, CNN-LSTM is 21 77 MB, a 25% improvement over the previous EM-CNN and other algorithms. The new proposed model CNN-LSTM is significantly smaller, simpler, and requires less storage than the last different model.

Table 6. The overall speed, drowsiness detection time, and compression of a different deep learning model.

Parameters	Workstation environment	EM-CNN	VGG-16	GoogLe-Net	Alex-Net	ResNet-50	Proposed CNN-LSTM
Compression (MB)	Lenovo workstation	33.6	2134	1265	1998	984	21.77
Drowsiness Detection Time (Second)	Lenovo workstation	66.7	88.34	96.65	56.87	89.90	26.88
Overall speed(fps)	Lenovo workstation	12.4	12.1	14.67	28	15.67	11.6

Table 6 shows the Overall speed, Drowsiness Detection Time, and Compression of a different deep learning model for driver drowsiness by model size, GPU memory, Lenovo workstation, and RealMe X7 Max. This study focused on estimating driver tiredness using videos recorded while the driver was on the road. We tested the proposed prediction models using an established dataset.

Table 6 shows that the model's accuracy under these circumstances was highly accurate. The films with "No Glasses," which depict the ideal road conditions, were the most

accurate. Sunglasses block the driver's vision and lower the quality of the characteristics the model could detect, and hence the classification accuracy was lowest. There are other features, such as the shape of the lips, the axis of the head, and so on, in addition to the eyes. Analytical analysis of the main characteristics that the CNN and LSTM models automatically transform into dynamic actions allows for a conclusion. All of the evaluation parameters were significantly improved by the suggested CNN-LSTM model. Therefore, the overall performance enhancement brought about by fusing the CNN model with the LSTM encourages its implementation in real-time applications. The method has an accuracy of 98.46%. The optimization objectives for loss functions in "Driver Drowsiness using CNN-LSTM and U-Net" involve setting up appropriate loss functions for binary classification (CNN-LSTM) and pixel-wise semantic segmentation (U-Net), and possibly combining these losses in a balanced manner when integrating the two models. Effective choice and tuning of these loss functions are critical for training a model that can accurately detect driver drowsiness based on visual cues. When working on the task of driver drowsiness detection using a combination of CNN-LSTM and U-Net, setting the appropriate loss functions is a crucial step in optimizing the neural network models. Loss functions quantify the error between predicted outputs and ground truth labels, and their choice impacts the training and performance of the models. The paper's contribution extends to the novel data acquisition methodology it employs. By capturing a range of facial movements and expressions – including eye closure duration, blinking patterns, and head orientation – the system acquires real-time data that is crucial for accurate drowsiness detection.

5.2. Limitations and Constraints

Binarization does not function for those with dark skin. Binarization methods often rely on contrast between foreground and background. Darker skin tones may have lower contrast under certain lighting conditions, making it challenging for standard binarization techniques to accurately separate features. Traditional binarization methods may not be sensitive to color variations in different skin tones. Grayscale-based methods may not capture the diversity of skin colors effectively. If the binarization model is trained on a dataset that lacks diversity in skin tones, it may not generalize well to individuals with dark skin. Addressing the issue of binarization not functioning well for individuals with dark skin requires a holistic approach involving technical improvements, data diversity, ethical considerations, and community engagement. It is essential to strive for fairness, inclusivity, and accuracy in image processing algorithms to avoid perpetuating biases and limitations. There cannot be any reflective materials behind the driver, another restriction. The system becomes more reliable as the background becomes more homogenous. A black sheet was placed behind the test participant to solve this issue for testing purposes. Rapid head movement was not permitted throughout testing. Since it can be compared to emulating a weary driver, this would be okay. Sometimes head motions are rarely lost by the system. The films that included "No Glasses," which depicted perfect driving circumstances, were the most accurate. The classification accuracy was lowest with sunglasses on, which blocked the driver's eyesight and reduced the quality of the traits the model could identify. In addition to the eyes, there are additional characteristics, such as the lips' form and the head's axis. So analyzing the significant properties that the CNN and LSTM models automatically turn into dynamic actions enables conclusions to be drawn. This is obvious since the system's algorithm is fundamentally dependent on binarization. Another restriction is that no reflective materials can be used behind the driver. When the backdrop gets more homogeneous, the system becomes more dependable.

Implement anomaly detection techniques to identify instances where the model might struggle due to unusual conditions, such as extremely low light. Experiment with different hyperparameter settings, including learning rates, batch sizes, and model architectures, to optimize performance. Ensure our model can handle challenging situations like glare, reflections, or unusual headlight shapes in real scene.

The proposed work is geared toward analyzing sequential data with both spatial and temporal aspects, while the other approach is focused on human action recognition within untrimmed video data using subspace clustering and coding-based techniques.CNN-LSTM with U-Net" is designed for analyzing sequential data with both spatial and temporal aspects, while "Sequential Order-Aware Coding-Based Robust Subspace Clustering" focuses on clustering and coding techniques for human action recognition in untrimmed videos.

The proposed work is designed for analyzing sequential data with both spatial and temporal aspects, while "GAN-Siamese Network for Cross-Domain Vehicle Re-identification" focuses on domain adaptation and similarity learning for vehicle recognition in intelligent transport systems.CNN-LSTM with U-Net" is designed for analyzing sequential data with both spatial and temporal aspects, as well as image segmentation tasks. The proposed work is designed for analyzing sequential data with both spatial and temporal aspects, as well as image segmentation tasks. On the other hand, "Spatio-Temporal Feature Encoding for Traffic Accident Detection in VANET Environment" focuses on encoding and analyzing spatio-temporal patterns for traffic accident detection in vehicular communication networks. The proposed work relates to neural network architectures and their applications in computer vision and deep learning, while "An Efficient and Secure Identity-Based Signature System for Underwater Green Transport System" pertains to cryptography and secure communication within the context of underwater transportation systems. These concepts are distinct in terms of their nature, purpose, and application domains.

6. Conclusion

The current advancements in road safety measures have been considerably propelled by the integration of Internet of Things (IoT) technology with facial movement analysis for the purpose of autonomous driver sleepiness detection. Recently, the discipline of Deep Learning has resolved many key issues. This study discusses a methodology for the detection of driver drowsiness through the utilization of real-time monitoring. In order to detect driver tiredness, the present study has devised a Deep Learning model utilizing a CNN-Long Short-Term Memory network architecture. The dataset comprises a diverse range of attempted methodologies, such as EM-CNN, VGG-16, GoogleNet, AlexNet, ResNet50, and CNN-LSTM. Various methods have been utilized for classification purposes, and it is evident that the CNN-LSTM approach demonstrates superior performance compared to other deep-learning techniques. Moreover, many more testing will be required to produce accurate results on its performance with the portability feature so that it can be used further using hardware supplies. In the near future, it will be advantageous to mitigate the potential hazards associated with accidents resulting from driver drowsiness. Regarding future research, our model could potentially benefit from the incorporation of an attention module because improving the model performance of drowsy detection involves a combination of optimizing the model architecture, fine-tuning hyperparameters, addressing data-related challenges, and incorporating advanced techniques.. Attention modules in fact play an important role in the human vision perceptron: they can allocate the available resources to selectively focus on processing the salient part instead of the whole scene, capturing long-range feature interactions and boosting the representation capability for CNN. This addition would enhance the model's performance by allowing it to consider more nuanced features throughout the categorization process.

References

1. Raj, P., & Raman, A. C. (2017). The Internet of Things: Enabling technologies, platforms, and use cases. CRC press.
2. Petridou, E., & Moustaki, M. (2000). Human factors in the causation of road traffic crashes. European journal of epidemiology, 16, 819-826.
3. Petridou, E., & Moustaki, M. (2000). Human factors in the causation of road traffic crashes. European journal of epidemiology, 16, 819-826.
4. Keall, M. D., & Newstead, S. (2012). Analysis of factors that increase motorcycle rider risk compared to car driver risk. Accident Analysis & Prevention, 49, 23-29.

5. Jin, W., Deng, Y., Jiang, H., Xie, Q., Shen, W., & Han, W. (2018). Latent class analysis of accident risks in usage-based insurance: Evidence from Beijing. Accident Analysis Prevention, 115, 79-88.
6. Mawson, A. R., & Walley, E. K. (2014). Toward an effective long-term strategy for preventing motor vehicle crashes and injuries. International journal of environmental research and public health, 11(8), 8123-8136.
7. Hughes, D. (2018). Case Study on the Experience of Street Racing (Doctoral dissertation, Capella University).
8. Schreier, D. R., Banks, C., & Mathis, J. (2018). Driving simulators in the clinical assessment of fitness to drive in sleepy individuals: A systematic review. Sleep medicine reviews, 38, 86-100.
9. Ayashm, S., Chehel Amirani, M., Valizadeh, M. (2022). Analysis of ecg signal by using an fcn network for automatic diagnosis of obstructive sleep apnea. Circuits, Systems, and Signal Processing, 41(11), 6411-6426.
10. M. Awais, N. Badruddin, M. Drieberg, "A hybrid approach to detecting driver drowsiness utilizing physiological signals to improve system performance and wearability", Sensors, vol. 17.no. 9, pp. 1991, Aug. 2017.
11. Garcia, C. I., Grasso, F., Luchetta, A., Piccirilli, M. C., Paolucci, L., & Talluri, G. (2020). A comparison of power quality disturbance detection and classification methods using CNN, LSTM and CNN-LSTM. Applied sciences, 10(19), 6755.
12. B. Warwick, N. Symons, X. Chen, K. Xiong, "Detecting driver drowsiness using wireless wearables", Proc. 12th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS), pp. 585-588, Oct. 2015.
13. G. Zhenhai, L. DinhDat, H. Hongyu, Y. Ziwen, W. Xinyu, "Driver drowsiness detection based on time series analysis of steering wheel angular velocity", Proc. 9th Int. Conf. Measuring Technol. Mechatron. Automat. (ICMTMA), pp. 99-101, Jan. 2017.
14. Z. Li, S. E. Li, R. Li, B. Cheng, J. Shi, "Online detection of driver fatigue using steering wheel angles for real driving conditions", Sensors, vol. 17, no. 3, pp. 495, Mar. 2017.
15. M. Saradadevi, P. Bajaj, "Driver fatigue detection using mouth and yawning analysis", Int. J. Comput. Sci. Netw. Secur., vol. 8, pp. 183-188, Jun. 2008.
16. Ye, M., Zhang, W., Cao, P., & Liu, K. (2021). Driver fatigue detection based on residual channel attention network and head pose estimation. Applied Sciences, 11(19), 9195.
17. I. Teyeb, O. Jemai, M. Zaied, C. B. Amar, "A novel approach for drowsy driver detection using Head posture estimation and eyes recognition system based on wavelet network", Proc. 5th Int. Conf. Inf. Intell. Syst. Appl. (IISA), pp. 379-384, Jul. 2014.
18. Bakker, B., Zabłocki, B., Baker, A., Riethmeister, V., Marx, B., Iyer, G., Anund, A. and Ahlström, C., 2021. A multi-stage, multi-feature machine learning approach to detect driver sleepiness in naturalistic road driving conditions. IEEE Transactions on Intelligent Transportation Systems.
19. Balam, V.P., Sameer, V.U. and Chinara, S., 2021. Automated classification system for drowsiness detection using convolutional neural network and electroencephalogram. IET Intelligent Transport Systems, 15(4), pp.514-524.
20. Chaabene, S., Bouaziz, B., Boudaya, A., Hökelmann, A., Ammar, A. and Chaari, L., 2021. Convolutional neural network for drowsiness detection using EEG signals. Sensors, 21(5), p.1734.
21. Dua, M., Singla, R., Raj, S. and Jangra, A., 2021. Deep CNN models-based ensemble approach to driver drowsiness detection. Neural Computing and Applications, 33(8), pp.3155-3168.
22. Sheykhivand, S., Rezaii, T. Y., Mousavi, Z., Meshgini, S., Makouei, S., Farzamnia, A.,& Teo Tze Kin, K. (2022). Automatic detection of driver fatigue based on EEG signals using a developed deep neural network. Electronics, 11(14), 2169.
23. Al-Hussein, W. A., Por, L. Y., Kiah, M. L. M., & Zaidan, B. B. (2022). Driver behavior profiling and recognition using deep-learning methods: In accordance with traffic regulations and experts guidelines. International journal of environmental research and public health, 19(3), 1470.
24. Arefnezhad, S., Hamet, J., Eichberger, A., Frühwirth, M., Ischebeck, A., Koglbauer, I. V., & Yousefi, A. (2022). Driver drowsiness estimation using EEG signals with a dynamical encoder–decoder modeling framework. Scientific reports, 12(1), 2650.
25. Jamshidi, S., Azmi, R., Sharghi, M. and Soryani, M., 2021. Hierarchical deep neural networks to detect driver drowsiness. Multimedia Tools and Applications, 80(10), pp.16045-16058.
26. Liu, P., Chi, H.L., Li, X. and Guo, J., 2021. Effects of dataset characteristics on the performance of fatigue detection for crane operators using hybrid deep neural networks. Automation in Construction, 132, p.103901.
27. Abbas, Q., & Alsheddy, A. (2020). Driver fatigue detection systems using multi-sensors, smartphone, and cloud-based computing platforms: a comparative analysis. Sensors, 21(1), 56.
28. Cui, J., Lan, Z., Zheng, T., Liu, Y., Sourina, O., Wang, L. and Müller-Wittig, W., 2021, September. Subject-Independent Drowsiness Recognition from Single-Channel EEG with an Interpretable CNN-LSTM model. In 2021 International Conference on Cyberworlds (CW) (pp. 201-208). IEEE.
29. Ye, M., Zhang, W., Cao, P. and Liu, K., 2021. Driver Fatigue Detection Based on Residual Channel Attention Network and Head Pose Estimation. Applied Sciences, 11(19), p.9195.
30. Ulrich, L., Nonis, F., Vezzetti, E., Moos, S., Caruso, G., Shi, Y., Marcolin, F. (2021). Can ADAS Distract Driver's Attention? An RGB-D Camera and Deep Learning-Based Analysis. Applied Sciences, 11(24), 11587.
31. Ulrich, L., Nonis, F., Vezzetti, E., Moos, S., Caruso, G., Shi, Y., Marcolin, F. (2021). Can ADAS Distract Driver's Attention? An RGB-D Camera and Deep Learning-Based Analysis. Applied Sciences, 11(24), 11587.
32. Bhuvaneswari, A., Thomas, J. T. J., & Kesavan, P. (2019). Embedded bi-directional GRU and LSTMLearning models to predict disasterson twitter data. Procedia Computer Science, 165, 511-516.

33. You, J., Jiang, D., Ma, Y. and Wang, Y., 2021. SpindleU-Net: An Adaptive U-Net Framework for Sleep Spindle Detection in Single-Channel EEG. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 29, pp.1614-1623.
34. Wu, E.Q., Xiong, P., Tang, Z.R., Li, G.J., Song, A. and Zhu, L.M., 2021. Detecting dynamic behavior of brain fatigue through 3-d-CNN-LSTM. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 52(1), pp.90-100.
35. Gupta, M., Thakur, N., Bansal, D., Chaudhary, G., Davaasambuu, B., & Hua, Q. (2022). CNN-LSTM hybrid real-time IoT-based cognitive approaches for ISLR with WebRTC: auditory impaired assistive technology. Journal of healthcare engineering, 2022.
36. Islam, M.Z., Islam, M.M. and Asraf, A., 2020. A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. Informatics in medicine unlocked, 20, p.100412.
37. Li, G., Lee, B. L., & Chung, W. Y. (2015). Smartwatch-based wearable EEG system for driver drowsiness detection. IEEE Sensors Journal, 15(12), 7169-7180.
38. Pauly, L., & Sankar, D. (2015, November). Detection of drowsiness based on HOG features and SVM classifiers. In 2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN) (pp. 181-186). IEEE.
39. Rahman, A., Islam, M. J., Band, S. S., Muhammad, G., Hasan, K., Tiwari, P. (2023). Towards a blockchain-SDN-based secure architecture for cloud computing in smart industrial IoT. Digital Communications and Networks, 9(2), 411-421.
40. Wang, W., Chen, Q., Yin, Z., Srivastava, G., Gadekallu, T. R., Alsolami, F., Su, C. (2021). Blockchain and PUF-based lightweight authentication protocol for wireless medical sensor networks. IEEE Internet of Things Journal, 9(11), 8883-8891.
41. Zhou, Z., Ding, C., Li, J., Mohammadi, E., Liu, G., Yang, Y., Wu, Q. J. (2022). Sequential Order-Aware Coding-Based Robust Subspace Clustering for Human Action Recognition in Untrimmed Videos. IEEE Transactions on Image Processing, 32, 13-28.
42. Zhou, Z., Li, Y., Li, J., Yu, K., Kou, G., Wang, M., Gupta, B. B. (2022). Gan-siamese network for cross-domain vehicle re-identification in intelligent transport systems. IEEE Transactions on Network Science and Engineering.
43. Zhou, Z., Dong, X., Li, Z., Yu, K., Ding, C., Yang, Y. (2022). Spatio-temporal feature encoding for traffic accident detection in VANET environment. IEEE Transactions on Intelligent Transportation Systems, 23(10), 19772-19781.
44. Zhou, Z., Gupta, B. B., Gaurav, A., Li, Y., Lytras, M. D., Nedjah, N. (2022). An efficient and secure identity-based signature system for underwater green transport system. IEEE Transactions on Intelligent Transportation Systems, 23(9), 16161-16169.
45. Flores, M. J., Armingol, J. M., & de la Escalera, A. (2010). Real-time warning system for driver drowsiness detection using visual information. Journal of Intelligent & Robotic Systems, 59(2), 103-125.
46. Manu, B. N. (2016, November). Facial features monitoring for real time drowsiness detection. In 2016 12th International Conference on Innovations in information technology (IIT) (pp. 1-4). IEEE.
47. Rahman, A., Sirshar, M., & Khan, A. (2015, December). Real time drowsiness detection using eye blink monitoring. In 2015 National software engineering conference (NSEC) (pp. 1-7). IEEE.
48. Anjali, K. U., Thampi, A. K., Vijayaraman, A., Francis, M. F., James, N. J., & Rajan, B. K. (2016, March). Real-time nonintrusive monitoring and detection of eye blinking in view of accident prevention due to drowsiness. In 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT) (pp. 1-6). IEEE.
49. Coetzer, R. C., & Hancke, G. P. (2011, June). Eye detection for a real-time vehicle driver fatigue monitoring system. In 2011 IEEE Intelligent Vehicles Symposium (IV) (pp. 66-71). IEEE.
50. Abtahi, S., Hariri, B., & Shirmohammadi, S. (2011, May). Driver drowsiness monitoring based on yawning detection. In 2011 IEEE International Instrumentation and Measurement Technology Conference (pp. 1-4). IEEE
51. Punitha, A., Geetha, M. K., & Sivaprakash, A. (2014, March). Driver fatigue monitoring system based on eye state analysis. In 2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014] (pp. 1405-1408). IEEE.
52. AL-Anizy, G. J., Nordin, M. J., & Razooq, M. M. (2015). Automatic driver drowsiness detection using haar algorithm and support vector machine techniques. Asian J. Appl. Sci, 8(2), 149-157.
53. Zhang, F., Su, J., Geng, L., & Xiao, Z. (2017, February). Driver fatigue detection based on eye state recognition. In 2017 International Conference on Machine Vision and Information Technology (CMVIT) (pp. 105-110). IEEE.
54. Reddy, B., Kim, Y. H., Yun, S., Seo, C., & Jang, J. (2017). Real-time driver drowsiness detection for embedded system using model compression of deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 121-128).
55. George, A., & Routray, A. (2016, June). Real-time eye gaze direction classification using convolutional neural network. In 2016 International Conference on Signal Processing and Communications (SPCOM) (pp. 1-5). IEEE.
56. Zhang, B., Wang, W., & Cheng, B. (2015). Driver eye state classification based on cooccurrence matrix of oriented gradients. Advances in Mechanical Engineering, 7(2), 707106.
57. Zhao, Z., Zhou, N., Zhang, L., Yan, H., Xu, Y., & Zhang, Z. (2020). Driver fatigue detection based on convolutional neural networks using EM-CNN. Computational intelligence and neuroscience, 2020.
58. Reddy, B., Kim, Y. H., Yun, S., Seo, C., & Jang, J. (2017). Real-time driver drowsiness detection for embedded system using model compression of deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 121-128).
59. Zhao, Z., Zhou, N., Zhang, L., Yan, H., Xu, Y., & Zhang, Z. (2020). Driver fatigue detection based on convolutional neural networks using EM-CNN. Computational intelligence and neuroscience, 2020.

60. Dua, M., Singla, R., Raj, S., & Jangra, A. (2021). Deep CNN models-based ensemble approach to driver drowsiness detection. Neural Computing and Applications, 33(8), 3155-3168.
61. Bekhouche, S. E., Ruichek, Y., & Dornaika, F. (2022). Driver drowsiness detection in video sequences using hybrid selection of deep features. Knowledge-Based Systems, 109436.
62. Gao, D., Wang, H., Guo, X., Wang, L., Gui, G., Wang, W., & He, T. (2023). Federated Learning Based on CTC for Heterogeneous Internet of Things. IEEE Internet of Things Journal.
63. Tiwari, P., Lakhan, A., Jhaveri, R. H., & Gronli, T. M. (2023). Consumer-centric internet of medical things for cyborg applications based on federated reinforcement learning. IEEE Transactions on Consumer Electronics.
64. Deng, D., Li, J., Jhaveri, R. H., Tiwari, P., Ijaz, M. F., Ou, J., & Fan, C. (2022). Reinforcement-Learning-Based Optimization on Energy Efficiency in UAV Networks for IoT. IEEE Internet of Things Journal, 10(3), 2767-2775.
65. Wang, Y., Wu, H., Jhaveri, R. H., & Djenouri, Y. (2023). DRL-Based URLLC-Constraint and Energy-Efficient Task Offloading for Internet of Health Things. IEEE Journal of Biomedical and Health Informatics.
66. Mungra, D., Agrawal, A., Sharma, P., Tanwar, S., & Obaidat, M. S. (2020). PRATIT: a CNN-based emotion recognition system using histogram equalization and data augmentation. Multimedia Tools and Applications, 79, 2285-2307.
67. Sarkar, J. L., Ramasamy, V., Majumder, A., Pati, B., Panigrahi, C. R., Wang, W., & Dev, K. (2022). I-Health: SDN-based fog architecture for IIoT applications in healthcare. IEEE/ACM Transactions on Computational Biology and Bioinformatics.
68. Weng, Y., & Lin, F. (2022). Multimodal emotion recognition algorithm for artificial intelligence information system. Wireless Communications and Mobile Computing, 2022.
69. Khajehali, N., Yan, J., Chow, Y. W., & Fahmideh, M. (2023). A Comprehensive Overview of IoT-Based Federated Learning: Focusing on Client Selection Methods. Sensors, 23(16), 7235.
70. Lea, C., Flynn, M. D., Vidal, R., Reiter, A., & Hager, G. D. (2017). Temporal convolutional networks for action segmentation and detection. In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 156-165).
71. Miraz, M. H. (2020). Blockchain of things (BCoT): The fusion of blockchain and IoT technologies (pp. 141-159). Springer Singapore.
72. Li, T. H. S., Kuo, P. H., Tsai, T. N., & Luan, P. C. (2019). CNN and LSTM based facial expression analysis model for a humanoid robot. IEEE Access, 7, 93998-94011.
73. Li, J., Jin, K., Zhou, D., Kubota, N., & Ju, Z. (2020). Attention mechanism-based CNN for facial expression recognition. Neurocomputing, 411, 340-350.
74. Anand, V., Gupta, S., Koundal, D., Nayak, S. R., Barsocchi, P., & Bhoi, A. K. (2022). Modified U-net architecture for segmentation of skin lesion. Sensors, 22(3), 867.
75. Wang, Q., Jia, K., & Liu, P. (2015, September). Design and implementation of remote facial expression recognition surveillance system based on PCA and KNN algorithms. In 2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP) (pp. 314-317). IEEE.
76. J. L. Sarkar et al., "I-Health: SDN-Based Fog Architecture for IIoT Applications in Healthcare," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2022, doi: 10.1109/TCBB.2022.3193918.
77. W. Wang et al., "Blockchain and PUF-Based Lightweight Authentication Protocol for Wireless Medical Sensor Networks," in IEEE Internet of Things Journal, vol. 9, no. 11, pp. 8883-8891, 1 June1, 2022, doi: 10.1109/JIOT.2021.3117762.