[1]Yalakala Dinesh Kumar

[2] Dr. Sanjay Kumar

Early Detection of Driver Drowsiness Detection using Automated Deep Artificial Intelligence Learning (ADAI)

Journal of Electrical Systems

Abstract: - Early detection of driver drowsiness significantly contributes to road accidents worldwide. This study presents a novel Driver Drowsiness Detector (DDD) system powered by Automated Deep Artificial Intelligence Learning (ADAI) to address this issue and improve road safety. Created by merging open pose and Gaze Tracking Networks (GTN), the Driver Drowsiness Detector system provides a comprehensive method for drowsiness identification. Sensor-based modules collect data from various vehicle sensors, such as steering wheel movements and accelerator/brake pedal dynamics. In contrast, vision-based modules use a front-facing camera-based monitoring system for the driver's facial expression, different types of eye movements, and head posture. The pre-trained model, such as VGG-19, can be fine-tuned for specific drowsiness detection tasks to leverage the knowledge learned from large-scale datasets. Diverse driving scenarios encompass lighting conditions, weather conditions, and driver demographics. The dataset is labeled with alert and drowsy states to enable supervised learning. The proposed DDD model learns to extract meaningful features from visual and sensor data, allowing it to detect drowsiness in real time. When signs of drowsiness are detected, the DDD system can provide real-time alerts to drivers, assisting them in remaining alert and focused on the road. This system contributes to a safer road environment and a reduction in road accidents by reducing the occurrence of drowsy driving incidents.

Keywords: Automated Deep Artificial Intelligence Learning (ADAI), Driver Drowsiness Detector (DDD), Gaze Tracking Networks (GTN), VGG-19.

I. INTRODUCTION

To reduce auto accidents, automotive safety equipment must be able to detect when drivers are sleepy. Every day, many individuals drive to and from work to enhance their comfort and standard of life or to get somewhere fast [1]. Due to this trend, heavy traffic is seen on highways and in urban areas. Conversely, one of the main reasons for auto accidents is drowsy driving. Two strategies can be used to prevent accidents: setting off alerts and identifying drowsy drivers early on. Additionally, more than 1.3 million people lose their lives in automobile accidents cases are increasing every year [2]. One of the leading causes of accidents is drivers' sleep deprivation. Reducing road accidents requires the use of technology in driver tiredness detection systems. There is much interest in detecting sleepy drivers and employing cameras, sensors, and other instruments to alert people about and avert deadly collisions. Automakers that employ driver assistance technologies include Mercedes-Benz, Tesla, and others [3]. These developments have helped drivers steer clear of accidents. Recently, Samsung and Eyesight worked together to use facial patterns and feature analyses to measure a driver's level of attention. Among their advances were variable cruise control, lane departure warnings, automatic braking, and assisted steering. The advancement of this methodology is a significant concern for the business, educational, and scientific communities. Deep learning (DL) for driver drowsiness detection is a novel application of artificial intelligence (AI) and computer vision techniques to improve road safety [4]. Deep neural networks (DNN) are used in this technology to screen a driver's level of alertness and to detect the signs of drowsiness or fatigue in real-time applications. The main goal is to prevent accidents caused by driver inattention and sleepiness, which significantly contribute to road accidents worldwide. Drowsy driving is a serious problem that endangers drivers and other road users. It can cause slow reaction times, poor decision-making, and catastrophic accidents in the worst-case scenario [5]. Traditional methods of combating drowsy driving, such as taking coffee breaks or opening windows, are ineffective. Deep learning technology has emerged as a powerful tool in developing automated and intelligent systems capable of monitoring driver behavior and intervening when necessary to address this critical safety concern.

[1]Research Scholar at the Department of Computer Science and Engineering, Kalinga University, Naya Raipur, Chattisgarh, India

[2]Associate Professor at the Department of Computer Science and Engineering, Kalinga University, Naya Raipur, Chattisgarh, India

E-mail: id: dinesh.bujji@gmail.com

In this paper, an Automated Deep Artificial Intelligence Learning (ADAI) was introduced to take measures for detecting the driver's drowsiness based on the eye state of the driver. An advanced pre-trained model and VGG-19 architecture are used to train the drowsiness eyes of the drivers. The MRL-Eye-dataset is used for experimental analysis, and it contains a training set with 40.4k images of closed eyes and 41.3k images of open eyes. For the testing model, 1500 images of closed eyes and 1500 images of open eyes were used for the experiment analysis. The dataset that is available for open source is https://www.kaggle.com/datasets/tauilabdelilah/mrl-eye-dataset.

II. Related Work

A novel approach to head position estimation was presented by Teyeb et al. [6]. The six potential head position models our system can identify are presented in the first part and further detailed in the second. Indeed, our approach may be divided into three main stages: First, we used two approaches that bare Viola and Jones's methods to detect the different types of driver's faces. Next, we remove the non-image and image reference coordinates from the face bounding box detection system. Finally, we measure the head inclination angle and the distances between the collected coordinates to classify the head state. Using a combination of the Internet of Things and deep learning advanced methods to enhance Long Short-Term Memory, VGG16 and InceptionV3, Suresh et al. [7] present a method to reliably and efficiently identify driver drowsiness. Transfer learning techniques are paired with several drowsiness signals to increase the performance and accuracy of the drowsiness detection system in different driving scenarios. Time-varying elements are also considered in the approaches derived from Long Short-Term Memory and DenseNet. Jetson Nano monitoring system receives a warning message and sounds from the IoT module when it detects driver weariness. The testing findings show that we can obtain a high accuracy of up to 98% with our deep neural network approach. Walizad et al. [8] suggest a behavioral identification approach based on a driver tiredness detection analysis to alert different data drivers about information before an accident situation and various locations. This methodology has the potential to save driver information, lives, and situation-based live updates and lessen traffic and accident information can be shared. This work trains the model using a convolutional neural network (CNN) to determine whether the driver's detection system is open or closed eyes. The collection includes photos taken from numerous MRL eye datasets. The suggested model uses computer vision techniques to handle the dataset's images before training, using different types of edge detection, grayscale conversion, and dilation. Then, real-time tracking of face landmarks from video frames is achieved using the Google location-based MediaPipe Face mesh model. The suggested trained model receives the retrieved, processed, and fed ocular region to make predictions. When the model notices that the driver is sleepy, it alerts them to take precautions. Compared to all other studies on sleepiness detection, the CNN model proposed and implemented in this study has an overall accuracy of 95%. A unique Internet of Things (IoT)-based solution was presented by Kongcharoen et al. [9] as open source. Another study assessed three driver's eye-based recognition methods that could be included in the open-source platforms to alert drivers when they start to lose consciousness. Three key components are included in this system: 1) Convolutional Neural Network model with Haar Cascade; 2) facial-based landmark points identification; 3) gaze detection using three types of face position identification. We test each combination of those factors one hundred times. The optimal method is selected based on the number of correct detections to detect blinking and closed eyes. After that, this algorithm is retested using the following factors: light (day and night), face angle (left, right, and center), camera angle (left and right), and glasses (on and off). Based on the data, the most accurate algorithm for recognizing a driver's eyes is CNN's Haar Cascade method, which has a 94% accuracy rate. Mandal et al. [10] have proposed a vision-based tiredness detection system for bus driver monitoring that is easy to deploy and can be adapted for usage in buses and large vehicles. The modules of the system comprise the following: head-shoulder detection, face detection, eye openness estimation, fusion, eye detection, and sleepiness measure percentage of eyelid closure (PERCLOS) estimation. The primary imaginative strategies are: There are two approaches to estimate eye states: 1) a fusion technique based on adaptive integration on multi-model detections of both eyes and 2) a spectral regression-based approach to estimating the continuous level of eye openness. The driver states are classified using a robust metric of PERCLOS on the continuous eye-opening level. Zhang et al. [11] present a semi-supervised method based on K-nearest neighbor (K-NN)-based ensemble learning for categorizing the circling vehicles' maneuvering patterns. Three components make up the framework: online classification, online model update, and initial model training. Using k-means clustering of the maneuvering behavior, cluster features are computed, and a set of micro-clusters is obtained to build the first model. Second, the incoming examples are classified using the ensemble K-NN-based learning technique. Lastly, an exponential decay function and error-driven representative learning update the model online. The suggested method's performance is verified using representative lane-changing and turning the algorithm. Budak et al. proposed that the three initial primary components of the sleepiness detection analysis are used to detect sleeping times [12]. The spectrograms that correlate to the raw EEG signals are used in the proposed building blocks of signals. Spectral entropy and instantaneous frequency features are derived from EEG spectrogram images. In contrast, the initial building block's energy and zero-crossing distribution features are computed from raw EEG signals. Pre-trained VGGNet and AlexNet are directly used in the second construction component to extract deep features from the EEG spectrogram images. The final building block uses the tunable Q-factor wavelet transform (TQWT) to divide the EEG signals into related sub-bands. The acquired sub-band spectrogram images are then processed to extract statistical characteristics, like the instantaneous frequency mean and standard deviation. An LSTM network receives each feature group from every construction block for classification. Using a variety of physiological signs, Papakostas et al. [13] examined how well they could distinguish between inattentive and sleepy driving. Four physiological signs are being examined in different stages: skin temperature, respiration, and blood volume pulse. In a simulated driving environment, data were gathered from 45 individuals at various times of the day while engaging in various physical and cognitive distractions. Chai et al.'s [14] introduction used a driving simulator to gather eleven steering-wheel-related characteristics. Variance analysis revealed that four variables significantly correlated with the driver's state. Three models were constructed based on the parameters chosen: an SVM model, a BP-NN model, and a MOL model. The MOL model's recognition accuracy was significantly greater than the two other models under the same classification settings. To address the aforementioned issues, a novel method of yawning detection based on subtle facial action identification is suggested in this paper by Yang et al. [15]. A 3D deep learning network with low-time sampling characteristics is presented to recognize delicate facial actions. This network employs SoftMax for classification and 3D convolutional and bidirectional long short-term memory networks to extract spatiotemporal features. A keyframe selection method has been developed to choose the most representative frame sequence from minor facial actions. This approach uses low-cost picture histograms to remove unnecessary frames quickly and uses the median absolute deviation to identify outliers. According to Deng et al. [16], a system known as DriCare uses video image classification and identifies the drivers' level of weariness based on behaviors like yawning, blinking, and length of eye closure without requiring them to wear any kind of wearable technology. We provide a novel face-tracking algorithm to increase tracking accuracy due to the flaws in earlier techniques. Additionally, we created a brand-new facial region identification technique based on 68 crucial features. Next, we assess the drivers' condition using these facial regions. According to Sparrow et al. [17], sleepiness-related commercial motor vehicle (CMV) crashes can be decreased by promptly detecting tiredness in CMV operations. This applies to driving manually and, ironically, becomes even more relevant as driving automation rises. The validity, efficacy, usefulness, and reliability of the sleepiness detection measures that are now available differ. An algorithm for detecting fatigued driving based on facial multi-feature fusion incorporating driver attributes was presented by Li et al. [18]. We initially design an upgraded YOLOv3-tiny convolutional neural network in order to capture the facial regions under complex driving conditions and remove the biases and inaccuracies caused by artificial feature extraction. Secondly, we offer the Eye Feature Vector (EFV) and Mouth Feature Vector (MFV), which describe the assessment parameters for the driver's mouth and eyes based on the Dlib toolkit. .. To generate the driver identity information library, offline training develops the driver biometric library, driver mouth state classifier library, and driver's eye state classifier library. Finally, we develop the driver identity verification and driver fatigue evaluation models using online tests. You and your colleagues proposed a real-time technique for detecting driving fatigue that considers each driver's unique characteristics [19]. A deep cascaded CNN was constructed to detect the facial region, eliminating the low accuracy problem caused by artificial feature extraction. The Dlib toolbox locates the frontal driver's facial landmarks in a frame. Based on the eye landmarks, a new parameter called the Eyes Aspect Ratio is introduced to determine the driver's level of fatigue in the current frame. The proposed method consists of two modules that consider the driver's eye size variations: online monitoring and offline training. In the first module, a novel SVM-based tiredness state classifier was learned using the eyes aspect ratio as an input. The second module then uses the learned classifier to track the online driver's condition.

A. Dataset Description

This dataset is merely a portion of the extensive collection of human eye images known as The MRL Eye Dataset. It's ready for jobs involving classification. This collection includes low- and high-resolution infrared photos taken with different equipment and under varying lighting circumstances. The dataset can be used to test trainable classifiers or multiple features. The images are categorized into multiple groups to facilitate algorithm comparison and to make them appropriate for classifier training and testing. Three distinct sensors' worth of images—the Intel RealSense RS 300 sensor, which has a resolution of 640 x 480; the IDS Imaging sensor, which has a resolution of 1280 × 1024; and the Aptina sensor, which has a resolution of 752 x 480—are included in the collection.

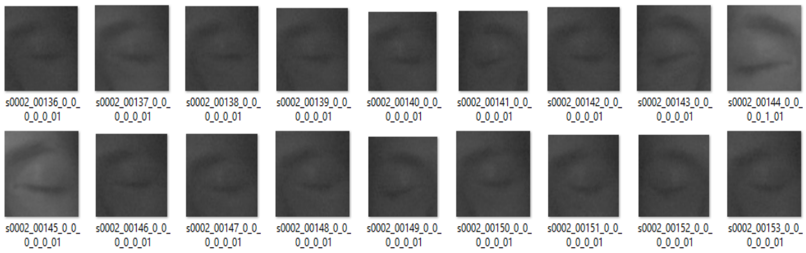

Figure 1: Dataset Closed Eyes Images

Figure 2: Dataset Open Eyes Images

Table 1: Total Dataset Samples Used for Experiments

	Open Eyes	Closed Eyes
Training Set	41292	40380
Testing Set	1657	1566

III VGG-19 PRE-TRAINED MODEL

Driver drowsiness detection is critical to road safety, aiming to prevent accidents caused by tired or distracted drivers. Deep learning and computer vision techniques are practical approaches to tackling this issue. Among the many pre-trained models available for various computer vision tasks, VGG-19 is a popular choice due to its strong performance in image classification tasks. The University of Oxford's Visual Geometry Group created the convolutional neural network architecture known as VGG-19, or Visual Geometry Group 19. It is well known for being easy to use and efficient at classifying images. As a member of the VGG family, which consists of several architectures at varying depths, VGG-19 is one of the deeper models with 19 layers.

The key features of the VGG-19 model include:

Deep Architecture:With 16 convolutional network layers and 3 fully connected layers, VGG-19 has 19 network layers in total. The network's depth makes it possible to photograph complex patterns and details.

Uniform Convolutional Layers: In VGG-19, the convolutional network layers are designed with small 3x3 filters and a stride of 1, ensuring the spatial resolution remains unchanged. This design choice contributes to the model's effectiveness.

Pooling Layers: Max-pooling layers with 2x2 filters reduce spatial dimensions and retain essential features while discarding less relevant information.

Pre-trained Weights: VGG-19 is often used with pre-trained weights using large-scale image datasets like ImageNet. These pre-trained weights capture a wide range of visual features and can be fine-tuned for specific tasks like driver drowsiness detection.

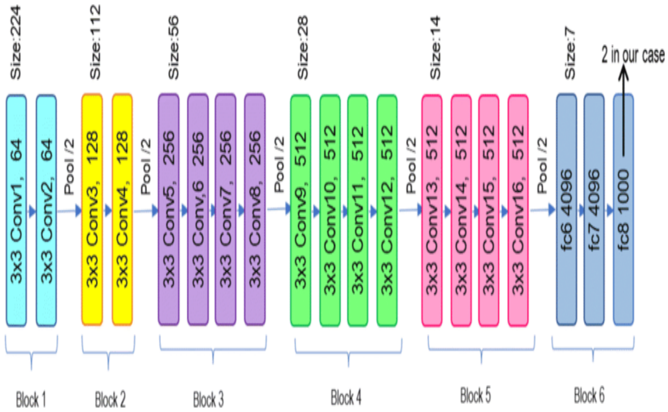

Figure 3: VGG-19 Architecture with 19 layers and 6 blocks

The following steps are used to apply the VGG-19 model to driver drowsiness detection:
Data Collection: The dataset of images or video frames containing alert and drowsy drivers. Annotate these images or frames to indicate the driver's state.
Data Preprocessing: Preprocess the dataset by resizing images to the input size expected by VGG-19 (typically 224x224 pixels) and normalizing pixel values.
Model Fine-tuning: Initialize the VGG-19 model with pre-trained weights and modify the final classification layer to have the desired output classes (alert or drowsy). You may also freeze some layers or use transfer learning techniques to adapt the model to your task.
Training: Train the modified VGG-19 model on your annotated dataset. Utilize techniques like data augmentation to improve model generalization.
Evaluation: The performance analysis uses accuracy, precision, recall, and F1-score metrics on a separate validation or test dataset.
Deployment: Once you have a trained model, you can deploy it in a real-world scenario, such as in a car's onboard system, to continuously monitor the driver's state and trigger alerts if drowsiness is detected.VGG-19's strong feature extraction capabilities make it a valuable tool for driver drowsiness detection when used with an appropriate dataset and fine-tuning strategy. It can help enhance road safety by providing early warnings to drivers and reducing the risk of accidents caused by drowsy driving.

A. Open Pose Technique for Driver Drowsiness Detection

OpenPose is a popular computer vision technique for estimating poses in images or videos, which involves tracking and identifying key points on a person's body. OpenPose isn't explicitly designed for this use case, but it can be integrated into a system by monitoring the driver's posture and facial expressions for signs of weariness.
This research uses OpenPose [20] to identify eighteen important human body locations. After receiving an image as input and output, the two-dimensional coordinates of essential locations. Even on straight roads, the driver must rotate the steering wheel slightly when driving regularly. This is because the driver continually adjusts the vehicle's state based on the conditions of the road. The vehicle's range lowers, and the operating speed decreases when the driver tires. As a result, the frequency and magnitude of variations may indicate the driver's level of weariness [21]. In addition to drowsy facial expressions such as nodding, yawning, and blinking, a fatigued driver will also have a reduced range and frequency of body movements. This study aims to determine the projected Euclidean distance of the arms, the area between the arms, and the dispersion degree of the wrist coordinate utilizing key points on the human body. These values will be used as fatigue features in this investigation of the relationship between driver fatigue and human posture. The features listed below demonstrate how important it is to identify driver drowsiness.

Eye Tracking: Monitoring eye movements, such as blink rate and gaze direction, can reveal drowsiness.

$$\text{BlinkRate} = \frac{\text{No of Blinks}}{\text{Time Interval}} \qquad (1)$$

Pupil Diameter: Pupil size can be measured as it tends to change with drowsiness.

$$\text{Pupil Diameter} = \frac{2 \times \text{radius}}{\text{Scale Factor}} \qquad (2)$$

Heart Rate Variability (HRV): Heart Rate Variability measures the variation in time between successive heartbeats.

$$\text{SDNN} - \text{Standard Deviation of NN intervals} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{(N} \text{NN}_i - \widehat{\text{NN}})^2} \qquad (3)$$

Electromyography (EMG): EMG measures muscle activity, which can change with drowsiness.

$$\text{EMG} = \int_{t_1}^{t_2} |x(t)|dt \qquad (4)$$

Behavioral Measures: Behavioral indicators like head nodding and steering wheel movement can be observed.

$$\begin{cases} \text{Alert} & \text{if Head Nodding Frequency} < \mathit{Threshold} \\ \text{Drowsy} & \text{if Head Nodding Frequency} \geq \text{Threshold} \end{cases} \qquad (5)$$

Thresholding: Decision thresholds are often used to classify drowsiness/alertness based on model outputs.

$$\begin{cases} \text{True} & \text{if } P(Y=1) \geq \text{Threshold} \\ \text{False} & \text{if } P(Y=1) < \mathit{Threshold} \end{cases} \qquad (6)$$

These factors and critical points represent various aspects of drowsiness detection, combining physiological, behavioral, and computational methods to provide a comprehensive approach to detecting drowsiness in real-world scenarios. Different applications may prioritize different indicators and techniques.

IV. Gaze Tracking Networks (GTN) for Driver Drowsiness Detection

Gaze Tracking Networks (GTN) are advanced technologies used in the automotive industry and driver assistance systems to improve road safety. This system uses computer vision and machine learning techniques to continuously monitor a driver's gaze, eye movements, and facial expressions to detect signs of drowsiness or distraction. GTN's primary goal is to prevent accidents by alerting drivers when they are not fully attentive or are on the verge of falling asleep behind the wheel. GTN uses eye-tracking technology to track a driver's eye movements accurately. This includes monitoring eye position, gaze direction, blink frequency, and pupil dilation. This data is critical in determining a driver's level of alertness. GTN algorithms analyze gaze-tracking data to detect signs of drowsiness or distraction. Prolonged periods of closed eyes or heavy eyelid drooping are common signs of drowsiness. These signals can activate a variety of warning systems. The GTN system can trigger various warning mechanisms when it detects drowsiness or distraction. These could include audible alerts, seat vibrations, visual warnings on the dashboard, or even autonomous vehicle interventions like slowing down or providing gentle steering inputs. The following steps are used to detect the face elevations: The viola-Jones algorithm uses Haar-like features and a cascade of classifiers. The basic idea is to compare the sum of pixel values in adjacent rectangular regions of an image to detect the presence of a face.

$$\text{ii}(a,b) = \sum [\text{Image}(p) \text{for all p such that p. a } \leq \text{a and p. b } \leq y] \qquad (7)$$

Where:

$\text{ii}(a,b)$is the value in the integral image at position (x, y).

Image (p) is the original image's pixel value at point p.

Haar-like Features: Haar-like features are rectangular filters that capture different facial characteristics. These features are applied to the integral image to compute the feature value quickly:

$$\text{Haar}(a,b) = \text{ii}(X) - \text{ii}(Y) - \text{ii}(K) + \text{ii}(Z) \qquad (8)$$

Where:

A, B, C, and D are the corners of the rectangular Haar-like feature.

Eye Localization: After detecting a face, the next step is frequently to localize the eyes within the detected face region. One common approach is using Haar-like features and cascade classifiers like face detection does. The specific Haar-like features aid in the detection of eyes, and the Viola-Jones algorithm is then applied within the detected face region. Equation (8) for Haar-like features and integral images is the same as that for face detection, as described above. Specific Haar-like features for eye detection are defined to capture eye characteristics such as the contrast between the iris and the sclera.

Pupil-Corneal Reflection (PCR) method

The Pupil-Corneal Reflection (PCR) method, also known as Purkinje image tracking or Purkinje-Sanson image tracking, is an eye-tracking technique used to measure gaze direction and eye movement. It captures and analyzes the reflection of light from the cornea and the anterior surface of the lens as it moves across the pupil (the Purkinje images). It can determine the gaze point and even estimate eye movement characteristics like saccades and smooth pursuit by tracking the positions of these reflections over time.

The following steps are involved in the PCR method:

Purkinje Images: There are typically four Purkinje images (PI-IV), but for the PCR method, we focus on the first two:

PI (Purkinje Image I): The reflection from the anterior surface of the cornea.

PII (Purkinje Image II): The reflection from the anterior surface of the lens.

Eye Model: The PCR method often uses a simplified eye model, treating the eye as a single refractive surface. This model simplifies calculations but may introduce some inaccuracies.

Cornea-Lens Separation: The distance between the anterior surfaces of the cornea and the lens is denoted as "d."

Pupil Coordinates: The position of the Purkinje images is defined in terms of coordinates within the pupil plane. These coordinates are often expressed as percentages of the pupil's diameter. PI and PII positions can be represented as (x1, y1) and (x2, y2), respectively.

Gaze Point: The gaze point on a display screen is represented as (X, Y) in screen coordinates. The relationship between the gaze point and the Purkinje image positions can be expressed as follows:

$$\mathrm{X} = \mathrm{a} * (\mathrm{x}_1 - \mathrm{x}_2) \quad (9)$$

$$\mathrm{Y} = \mathrm{a} * (\mathrm{y}_1 - \mathrm{y}_2) \quad (10)$$

Here, "a" is a constant that depends on the optical properties of the eye and the imaging system. It can be calculated based on the known parameters of the eye and the imaging setup.

Eye Movement Analysis: By tracking the changes in the positions of PI and PII over time, you can estimate various eye movements, such as saccades (rapid gaze shifts) and smooth pursuits (smooth tracking of a moving target).

Data Representation:

Gaze estimation typically involves sequences of eye images or facial landmarks captured over time. Each input data point can be represented as a sequence of feature vectors, denoted as:

Input Sequence: $\mathrm{A} = [\mathrm{a}_1, \mathrm{a}_2, \dots \dots \mathrm{a}_\mathrm{T}]$ (11)

Where T is the sequence length and each a_T represents an input feature vector (e.g., an image or a vector of facial landmarks) at time step t.

Target Gaze Directions: $\mathrm{B} = [\mathrm{b}_1, \mathrm{b}_2, \dots \dots \mathrm{b}_\mathrm{T}]$ (12)

Where each b_T represents the gaze direction (e.g., a 2D vector representing horizontal and vertical gaze angles) at time step t.

V. RNN ARCHITECTURE

RNNs process sequences of data step by step, maintaining hidden state information that captures information from previous time steps. The basic equations for a simple RNN cell are as follows:

Hidden State Update:

$$h_t = f(W_{hh} * h_{(t-1)} + W_{hx} * a_t) \quad (13)$$

Output at Time Step t:

$$b_t = g(W_{hy} * h_t) \quad (14)$$

Where:
h_t: Hidden state at time step t.
a_t: Input feature vector at time step t.
W_{hh}, W_{hx}, W_{hy}: Weight matrices for hidden state, input, and output connections.
f and g are activation functions (e.g., sigmoid or hyperbolic tangent) applied element-wise.

Training Objective:
The goal during training is to minimize the error between predicted and ground truth gaze directions. A standard loss function for gaze estimation is mean squared error (MSE):

$$\text{Loss} = {}^{(1}\!/_{T)} * \sum(||b_t - \hat{b}_t||^2), \text{for } t = 1 \text{ to } T \quad (15)$$

Where:
b_t Ground truth gaze direction at time step t.
$\hat{b}_t$ Predicted gaze direction at time step t.
T: Sequence length.
Back-propagation through Time (BPTT):
Training an RNN involves using back-propagation to update the network's weights. In the case of RNNs, this is often done through BPTT, an extension of the standard back-propagation algorithm.

A. Performance Metrics

The proposed model's performances were analyzed using several parameters that show a huge impact on output.

$$\text{Accuracy(ACC)} = \frac{\text{True Positive} + \text{True Negative}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$$

$$\text{Precision(Pre)} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Sensitivity (Sn)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

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

$$\text{F1} - \text{Score} = 2 * \frac{(P * R)}{(P + R)}$$

B. Results and Discussion
The experimental analysis used Python programming language with a high configuration system. The system hardware contains 16 GB RAM and 1 TB of hardware that helps to solve loading issues. Libraries such as Keras, Pandas, matplotlib, and other ML libraries help improve the proposed approach's performance.

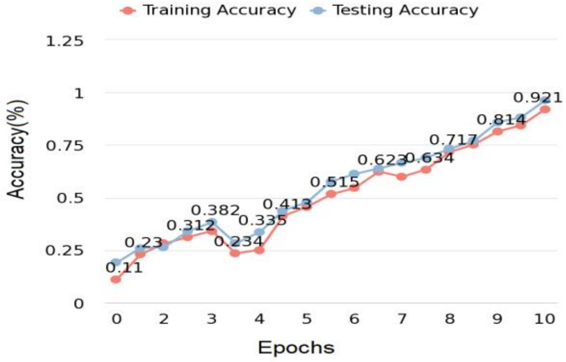

Figure 4: Training and Testing Accuracy

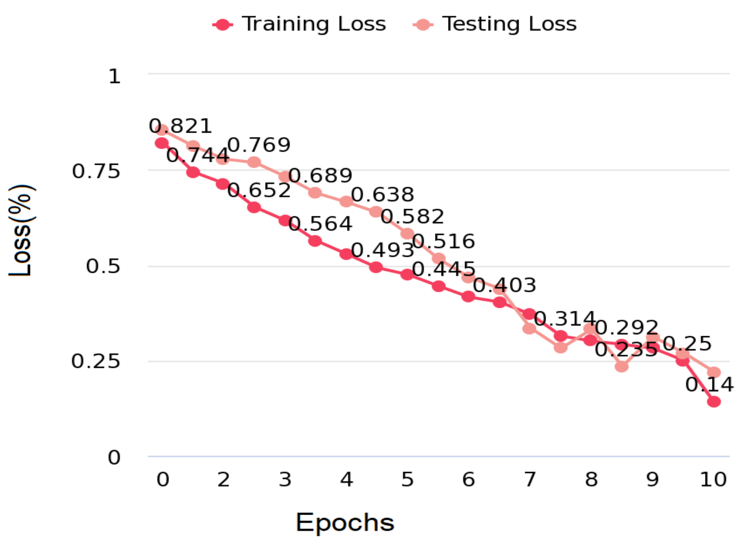

Figure 5: Training and Testing Loss

Table 2: The overall performances of Existing and Proposed Algorithms

Algorithms	Acc	Pre	Sn	Sp	F1-Score
Adaboost	83.34	84.12	80.12	79.34	80.45
LSTM	87.45	88.23	86.67	87.12	86.87
ADAI	93.56	94.23	95.42	96.87	96.56

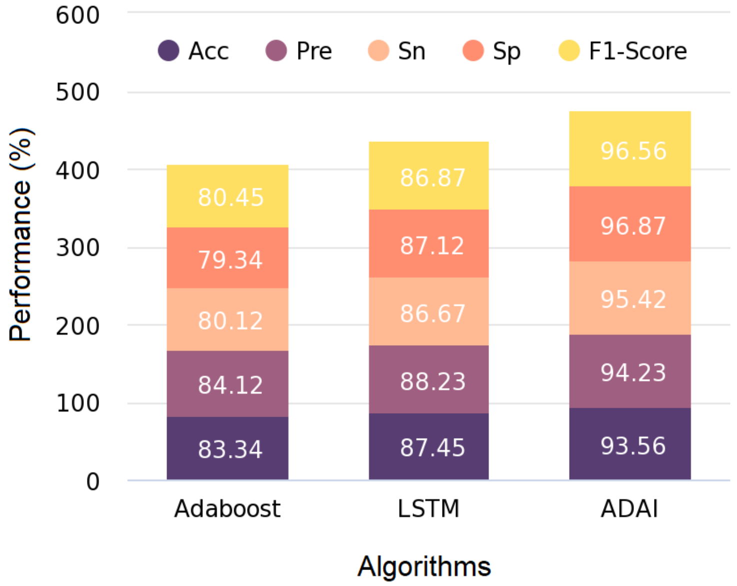

Figure 6. Comparison methods

VI. CONCLUSION

ADAI detection systems have the potential to significantly enhance road safety by identifying and alerting drowsy drivers, thus preventing accidents caused by driver fatigue. ADAI systems are pivotal in improving the system's road safety and security features. Drowsy driving is a significant cause of accidents worldwide, and early detection systems can help prevent accidents and reduce injuries, among others. ADAI systems continuously monitor driver behavior and facial cues to identify signs of drowsiness in real time. This proactive approach ensures that warnings are issued promptly, reducing the risk of accidents. AI algorithms used in ADAI systems are trained on vast datasets, enabling them to accurately recognize subtle signs of drowsiness, such as drooping eyelids, yawning, erratic steering, or slowed reaction times. ADAI systems can log driver behavior data, which can be helpful for post-incident analysis and understanding patterns of drowsiness among different demographics. Integrating ADAI into vehicles is becoming more commonplace, which means that this technology can benefit from the existing infrastructure of modern cars, such as sensors and communication systems. Finally, early detection of driver drowsiness using ADAI has the potential to save lives and reduce accidents on the road. While there are challenges to overcome, the continued advancement of AI and deep learning technologies and the features of each model can increase awareness of the dangers of drowsy driving, making ADAI a critical tool in improving road safety. It is an area of research and development that holds significant promise for the future.

ACKNOWLEDGMENTS

Our thanks to the experts who have contributed to the development of the template.

REFERENCES

[1] Betser, Michael. “Speaker Diarization Using Bottom-up Clustering Based on a Parameter-Derived Distance between Adapted GMMs.” Proc. ICSLP (2004):

[2] httS. Madikeri and H. Bourlard, "KL-HMM based speaker diarization system for meetings," 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 2015, pp. 4435-4439, doi: 10.1109/ICASSP.2015.7178809.

[3] Barhoush, M., Hallawa, A. & Schmeink, A. Speaker identification and localization using shuffled MFCC features and deep learning. Int J Speech Technol 26, 185–196 (2023). https://doi.org/10.1007/s10772-023-10023-2

[4] Nemer, Elias & Goubran, Rafik & Mahmoud, Samy. (2001). Robust voice activity detection using higher-order statistics in the LPC residual domain. Speech and Audio Processing, IEEE Transactions on. 9. 217 - 231. 10.1109/89.905996.

[5] Indu, D. ., & Srinivas, Y. . (2024). A Cluster-Based Speaker Diarization System Combined with Dimensionality Reduction Techniques . International Journal of Intelligent Systems and Applications in Engineering, 12(14s), 125–132.

[6] Daniel Garcia-Romero, David Snyder, Gregory Sell, Daniel Povey, and Alan McCree, “Speaker diarization using deep neural network embeddings,” in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017, pp. 4930–4934.

[7] Sepp Hochreiter and Jurgen Schmidhuber, “Long short-term ¨ memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] Ulrike Von Luxburg, “A tutorial on spectral clustering,” Statistics and computing, vol. 17, no. 4, pp. 395–416, 2007.

[9] Huazhong Ning, Ming Liu, Hao Tang, and Thomas S Huang, “A spectral clustering approach to speaker diarization.,” in INTERSPEECH, 2006.

[10] Philip Andrew Mansfield, Quan Wang, Carlton Downey, Li Wan, and Ignacio Lopez Moreno, “Links: A highdimensional online clustering method,” arXiv preprint arXiv:1801.10123, 2018.

[11] P. Kenny, “Bayesian speaker verification with heavy-tailed priors,” in Proc. Odyssey Speaker and Language Recognition Workshop, Brno, Czech Republic, Jul. 2010.

[12] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 4, pp. 788–798, 2011.

[13] D. Reynolds, T. Quoter, and R. Dunn, “Speaker verification using adapted Gaussian mixture models,” Digital Signal Processing, vol. 10, no. 1, pp. 19–41, 2000.

[14] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, “Joint factor analysis versus eigenchannels in speaker recognition,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 15, pp. 1435–1447, 2007.

[15] H. Aronowitz, “Text-dependent speaker verification using a small development set,” in Proc. Odyssey Speaker and Language Recognition Workshop, Singapore, Jun. 2012.

[16] T. Stafylakis, P. Kenny, P. Ouellet, P. Perez, J. Kockmann, and P. Dumouchel, “Text-dependent speaker recognition using PLDA with uncertainty propagation,” in Interspeech, Lyon, France, Aug. 2013.

[17] A. Larcher, K.-A. Lee, B. Ma, and H. Li, “Phoneticallyconstrained PLDA modeling for text-dependent speaker verification with multiple short utterances,” in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vancouver, Canada, May 2013.

[18] D. Garcia-Romero, X. Zhang, A. McCree, and D. Povey, "Improving speaker recognition performance in the domain adaptation challenge using deep neural networks," in IEEE Spoken Language Technology Workshop (SLT), South Lake Tahoie, NV, USA, Dec. 2014, pp. 378–383.

[19] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phoneticallyaware deep neural network," in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Florence, Italy, May 2014, pp. 1695–1699.

[20] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," IEEE Signal Processing Letters, 2005.

[21] E. Variani, X. Lei, E. McDermott, I. Lopez-Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Florence, Italy, May 2014.

[22] S. Parveen, A. Qadeer, and P. Green, "Speaker recognition with recurrent neural networks," in Sixth International Conference on Spoken Language Processing, ICSLP 2000 / INTERSPEECH 2000, Beijing, China, Oct 2000, pp. 16–20.

[23] J. Gonzalez-Dominguez, I. Lopez-Moreno, H. Sak, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic language identification using long short-term memory recurrent neural networks," in Interspeech, Singapore, Sep. 2014, pp. 2155–2159.