



Article

# Non-Invasive Fatigue Detection and Human–Machine Interaction Using LSTM and Multimodal AI: A Case Study

Muon Ha <sup>1,\*</sup>, Yulia Shichkina <sup>2</sup> and Xuan-Hien Nguyen <sup>3</sup>

<sup>1</sup> Faculty of Information Technology, Telecommunications University, Nha Trang 650000, Vietnam

<sup>2</sup> Faculty of Computer Science and Technology, Saint Petersburg Electrotechnical University, 197022 Saint Petersburg, Russia; strange.y@mail.ru

<sup>3</sup> School of Mechanical and Automotive Engineering, Hanoi University of Industry, Ha Noi 100000, Vietnam; hien.nguyen15@hau.edu.vn

\* Correspondence: muon.ha@mail.ru

**Abstract:** Fatigue in high-stress work environments poses significant risks to employee performance and safety. This study introduces a non-invasive fatigue detection system utilizing facial parameters processed via a Long Short-Term Memory (LSTM) neural network, coupled with a human–machine interaction interface via a Telegram chatbot. The system analyzes eye blink patterns and facial expression changes captured through a webcam, achieving an accuracy of 92.35% on the UTA-RLDD dataset. An interactive feedback mechanism allows users to verify predictions, enhancing system adaptability. We further propose a multimodal AI framework to integrate physiological and environmental data, laying the groundwork for broader applications. This approach provides an effective solution for early fatigue detection and adaptive collaboration between humans and machines in real-time settings.

**Keywords:** fatigue detection; human–machine interaction; LSTM; multimodal AI; non-invasive; Telegram chatbot



Academic Editors: Wei Liu, Jan Auernhammer and Takumi Ohashi

Received: 17 April 2025

Revised: 10 June 2025

Accepted: 11 June 2025

Published: 13 June 2025

**Citation:** Ha, M.; Shichkina, Y.; Nguyen, X.-H. Non-Invasive Fatigue Detection and Human–Machine Interaction Using LSTM and Multimodal AI: A Case Study. *Multimodal Technol. Interact.* **2025**, *9*, 63. <https://doi.org/10.3390/mti9060063>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Fatigue affects professions that demand sustained attention. These include driving, operating technical systems, and providing medical care. It reduces physical and mental performance. There are a lot of causes, including prolonged exertion, inadequate rest, or environmental stressors. In high-stakes environments, fatigue impairs reaction time, decision-making, and situational awareness. This leads to lower productivity and higher error risks. For example, fatigued drivers contribute to many road accidents [1]. Tired medical professionals may endanger patient safety [2]. Operators of complex systems, such as those in aerospace, face risks when fatigue delays responses to critical events [3]. Effective fatigue detection and mitigation are, thus, essential across multiple domains.

Traditional methods like electroencephalography (EEG) and electrooculography (EOG) measure physiological signals. EEG detects brain wave changes linked to drowsiness [4]. EOG tracks eye movements to identify blink patterns [5]. These approaches offer high precision. However, they require electrodes on the scalp or around the eyes. This setup causes discomfort and restricts mobility. Drivers or operators cannot use such systems daily. As a result, these methods lack practicality for widespread use.

Non-invasive alternatives use computer vision and artificial intelligence (AI). Cameras capture video data to analyze behavioral cues. Eye closure duration, measured as Percentage of Eye Closure (PERCLOS), indicates drowsiness [6]. Deep learning models,

like Convolutional Neural Networks (CNNs), detect yawning or head tilting [7]. These systems need no physical contact. They integrate into devices like dashboards or computers. However, current non-invasive methods focus only on fatigue identification. They lack dynamic interaction with users. This limits adaptation to individual differences, such as blinking habits, or contextual factors, like lighting conditions.

Existing systems also miss feedback mechanisms. Fatigue varies between individuals. Sleep history, stress, and environment affect it. Without user input, systems cannot adjust for these factors. For instance, frequent blinking may stem from dry eyes, not drowsiness [2]. A focused expression might resemble exhaustion. Unidirectional systems risk false predictions. This reduces trust and utility.

We propose a non-invasive fatigue detection system. It combines AI analysis with human-machine interaction. A Long Short-Term Memory (LSTM) network processes eye blinks and facial expressions from webcam video. LSTM models temporal patterns, such as blink duration changes, better than static methods. A Telegram chatbot enables real-time feedback. It asks users to confirm or clarify fatigue predictions. This improves accuracy and builds a personalized knowledge base. Our broader project, with Professors Shichkina and Krinkin [8], includes a multimodal AI framework. It integrates physiological data, like heart rate from smartwatches, and environmental factors, like temperature. This enhances robustness beyond facial cues alone.

Our goals are twofold. First, we aim to detect fatigue accurately in real time using vision-based techniques. Second, we seek to create an adaptive interaction mechanism. This blends human input with machine intelligence. The system targets safety and efficiency in high-stakes settings.

In this paper, the term “fatigue” is used in a broad sense that includes both cognitive fatigue—resulting from mental workload—and drowsiness—linked to sleepiness and reduced alertness. While these states differ physiologically, they often overlap in behavioral and biometric manifestations. This perspective aligns with prior multimodal studies and with the structure of datasets such as UTA-RLDD, where fatigue and drowsiness are not always strictly separated. We adopt this integrated usage throughout the paper.

Section 2 provides a comprehensive literature review of existing fatigue detection techniques, focusing on invasive and non-invasive approaches, datasets used in fatigue research, and current multimodal and interactive methods. Section 3 details the methodology, including data processing, LSTM modeling, and interaction design. Section 4 describes experimental methods. Section 5 discusses the results, limitations, and potential improvements. Finally, Section 6 concludes with key contributions and future research directions.

## 2. Literature Review

Fatigue detection research spans 15 years. It focuses on safety and productivity in high-stakes settings. Methods are divided into two groups. Invasive methods use physiological signals. Non-invasive methods use behavioral cues. Datasets support these studies. Multimodal and interactive approaches also emerge. This section reviews key works and identifies gaps.

### 2.1. Invasive Fatigue Detection Methods

Invasive methods measure physiological signals directly. Electroencephalography (EEG) and electrooculography (EOG) lead this group. Lal et al. [3] used EEG to detect drowsiness in drivers. They analyzed alpha and theta waves. Accuracy reached 85%. Jap et al. [9] extended EEG analysis with spectral components. Their system classified fatigue states precisely. Huo et al. [10] combined EEG and EOG. Their method achieved a correlation coefficient of 0.8080 and RMSE of 0.0712 in predicting fatigue levels from EEG

and forehead EOG signals, indicating reliable performance. Trejo et al. [11] applied EEG to estimate mental fatigue and achieved classification accuracies between 91.12% and 100% (mean = 98.3%) using machine learning on single 13-s EEG epochs. Vicente et al. [12] used heart rate variability (HRV) extracted from ECG to detect driver drowsiness. This approach reduced invasiveness but still required wearable sensors, achieving up to 93.1% precision (positive predictive value) and 85.3% sensitivity when trained with a merged dataset. These methods offer high precision. However, electrodes or sensors limit practicality. Users find them uncomfortable for daily tasks.

## 2.2. Non-Invasive Fatigue Detection Methods

Non-invasive methods rely on cameras, and Assari and Rahmati [13] proposed a real-time drowsiness detection system using facial expressions (eyes, mouth, eyebrows) with infrared imaging to ensure robustness under varying lighting conditions. Rahman et al. [14] proposed a real-time drowsiness detection system based on eye blink monitoring. Their webcam-based method achieved 94% accuracy under good lighting conditions. Flores et al. [15] built a driver warning system. It used Support Vector Machines (SVM) to classify visual cues. Gao et al. [16] linked facial expressions to stress. Their system reached 85% accuracy in cars. Ma et al. [17] applied two-stream CNNs to depth videos. They detected yawning with 90% accuracy. Chirra et al. [17] used deep CNNs for eye states. Results excelled in controlled settings. Shakeel et al. [18] combined the eyes and mouth features. Their real-time system worked well, but needed strong hardware. You et al. [19] developed CarSafe. This app used dual smartphone cameras. The accuracy was 83%. Mehta et al. [20] proposed Eye Aspect Ratio (EAR). It ran efficiently on basic devices. Soukupová and Čech [21] refined EAR for blinks. Their method suited real-time use. These approaches avoid physical contact; yet, they struggle with lighting or occlusion. Most lack adaptability to individual differences.

## 2.3. Datasets Supporting Fatigue Research

Datasets standardize fatigue studies. Ghoddosian et al. [22] created UTA-RLDD. It includes 180 videos from 60 subjects. Three states—normal, low fatigue, and fatigued—are recorded. Conditions vary from indoor to outdoor. This dataset fits temporal models. We use it in this study. Abtahi et al. [23] built YawDD. It features 30 subjects yawning. The angles and lighting differ. YawDD aids yawn detection but lacks fatigue labels. We exclude it here due to limited scope. These datasets improve testing. However, they miss contextual data like mood or environment.

## 2.4. Multimodal and Interactive Approaches

Multimodal systems combine data sources. Lin et al. [24] merged EEG with behavioral cues. Their wireless system detected drowsiness in real time. Ramzan et al. [25] reviewed hybrid methods. They suggest blending HRV and visual data. Shichkina and Krinkin [8] proposed a multimodal AI framework. It aggregates data and seeks human input in a published study. This inspires our work. Interactive systems are rare. Flores et al. [15] and You et al. [19] issue alerts only. They do not refine predictions with user feedback. This limits personalization.

## 2.5. Gaps and Our Contribution

Invasive methods excel in precision. Yet, they lack practicality [9–14]. Most focus solely on detection without considering adaptation to individual variability, such as blinking frequency or lighting conditions. Deep learning improves accuracy [17–19], but requires substantial computational resources. Many systems provide only fatigue scores, lacking user feedback integration [13,19].

Datasets like UTA-RLDD [22] support temporal modeling but lack multimodal annotations. We address this by applying LSTM to capture temporal fatigue patterns, enhancing static approaches used in [16,17]. In addition, we integrate a Telegram chatbot to incorporate real-time user feedback, addressing the adaptability gap identified in [13,19].

Our system builds on the multimodal AI vision proposed by Shichkina and Krinkin [8], fusing facial features, physiological signals, and contextual data to improve robustness and usability in real-world settings.

### 3. Proposed Method

This section presents a comprehensive methodology for non-invasive fatigue detection, designed to address the shortcomings of existing systems through a novel integration of temporal deep learning, real-time human–machine interaction, and a scalable multimodal framework. Our approach leverages advanced AI techniques to model fatigue dynamics, incorporates user feedback to enhance adaptability, and proposes a forward-looking extension for multi-source data fusion. Below, we detail the system architecture, data processing pipeline, fatigue assessment model, interaction mechanism, and multimodal extension, emphasizing scientific rigor, computational efficiency, and practical applicability.

#### 3.1. System Architecture

The system is structured as a modular pipeline comprising three core components, each optimized for specific tasks yet interconnected to form a cohesive fatigue detection framework (see Figure 1):

(1) Data Collection Module: Acquires real-time video streams of the user’s face via a standard webcam operating at 30 frames per second (fps), ensuring minimal hardware requirements while capturing sufficient temporal resolution.

(2) Fatigue Assessment Module: Employs a Long Short-Term Memory (LSTM) neural network to process sequential facial parameters, classifying fatigue states into three categories: normal, low fatigue, and fatigued. This module leverages the temporal nature of fatigue indicators, distinguishing it from static classification approaches.

(3) Human–Machine Interaction Module: Integrates a Telegram-based chatbot to solicit user feedback, enabling a closed-loop system that refines predictions and builds a dynamic knowledge base. This interactivity addresses the lack of adaptability in traditional systems.



**Figure 1.** Fatigue Detection System.

The architecture is designed for modularity and scalability, allowing independent optimization of each module (e.g., swapping LSTM for GRU or replacing Telegram with another platform) while maintaining a unified workflow. Unlike unidirectional systems, our feedback loop introduces a human-in-the-loop paradigm, enhancing both accuracy and user trust.

#### 3.2. Data Processing

Data processing converts raw video frames into a feature set optimized for temporal analysis, focusing on two key fatigue indicators: eye dynamics and facial expression changes. We employ the dlib library with a pre-trained shape predictor (68-point facial landmark model) to extract coordinates  $(x,y)$  for each frame, processed at 30 fps. The pipeline includes landmark detection, feature extraction, and normalization, detailed below.

### 3.2.1. Eye Aspect Ratio (EAR) and Blink Dynamics

Eye behavior is a primary fatigue marker, with drowsiness often manifesting as prolonged blinks or reduced eye openness. For each eye, six landmarks ( $P_1$  to  $P_6$ ) are extracted, following Soukupová and Čech [21]. The EAR is calculated as shown in Equation (1):

$$EAR = \frac{\|P_2 - P_6\| + \|P_3 - P_5\|}{2 \times \|P_1 - P_4\|}, \quad (1)$$

In (1):

- $P_1, P_4$ : Horizontal eye corners (left and right)
- $P_2, P_3, P_5, P_6$ : Vertical points on the upper and lower eyelids
- $\|P_i - P_j\| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$  is the Euclidean distance

The EAR normalizes vertical eye height by width, mitigating variations due to head tilt or camera distance. To capture temporal dynamics, we analyze a sequence of 10 consecutive blinks (approximately 10–15 s), extracting four features:

- Peak EAR ( $EAR_{max}$ ): Maximum value in a blink cycle, typically 0.3–0.4 when eyes are fully open
- Minimum EAR ( $EAR_{min}$ ): Minimum value during closure, approaching 0.05–0.1 for a full blink
- Blink Duration:

$$t_{blink} = (t_{end} - t_{start}) \times \frac{1}{2}, \quad (2)$$

- Blink Frequency:

$$f_{blink} = \frac{10}{\Delta t_{total}} \times 60, \quad (3)$$

Blink detection uses a threshold-based algorithm: a blink is identified when EAR drops below 0.2 (empirically determined) for at least three frames and rises above it, ensuring robustness against noise or partial closures. Figure 2 demonstrates the analysis of EAR and blink frequency over a 10-blink sequence.

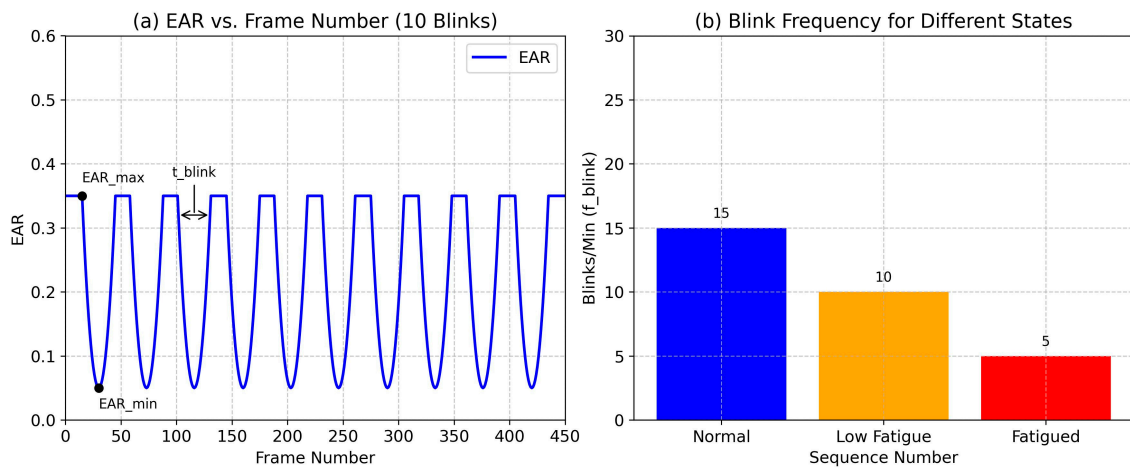


**Figure 2.** Analysis of Eye Aspect Ratio (EAR) and blink frequency for fatigue detection.

### 3.2.2. Facial Expression Index

Facial expressions provide contextual cues about fatigue, such as drooping features or reduced mobility. We compute Euclidean distances between key landmark pairs per frame:

Eye Distance:

$$d_{eyes,i} = \sqrt{(x_{37} - x_{46})^2 + (y_{37} - y_{46})^2}, \quad (4)$$

where points 37 and 46 are outer eye corners.

Mouth Distance:

$$d_{lip,i} = \sqrt{(x_{49} - x_{58})^2 + (y_{49} - y_{58})^2}, \quad (5)$$

where points 49 and 58 are vertical mouth extremes.

To quantify change over time, we normalize distances across the sequence:

$$\Delta d_{eyes,i} = \frac{d_{eyes,i} - d_{eyes,min}}{d_{eyes,max} - d_{eyes,min}}, \quad \Delta d_{lip,i} = \frac{d_{lip,i} - d_{lip,min}}{d_{lip,max} - d_{lip,min}}, \quad (6)$$

where  $d_{min}$  and  $d_{max}$  are the minimum and maximum distances over  $n$  frames (typically 300–450). The final feature,  $\Delta d$ , is the average rate of change between blinks:

$$\Delta d = \frac{1}{9} \sum_{k=1}^9 \sqrt{(\Delta d_{eyes,k+1} - \Delta d_{eyes,k})^2 + (\Delta d_{lip,k+1} - \Delta d_{lip,k})^2}, \quad (7)$$

This captures subtle shifts in facial tension, with higher values indicating fatigue-related relaxation. Figure 3 shows the mapping of key facial landmarks used to extract features such as eye and mouth distances.

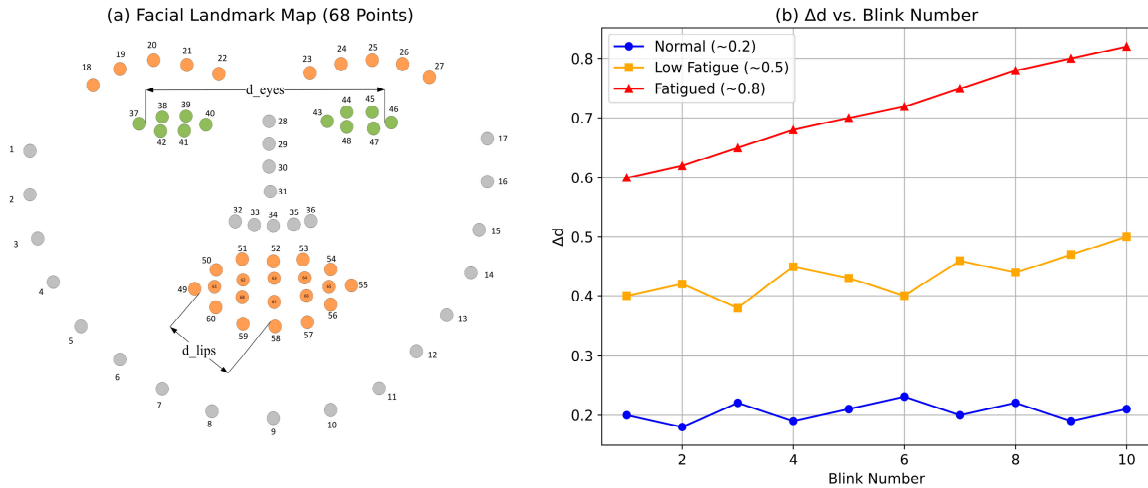


**Figure 3.** Facial landmark mapping and facial expression dynamics for fatigue detection.

### 3.2.3. Feature Preprocessing

Features are standardized to zero mean and unit variance:

$$x' = \frac{x - \mu}{\sigma}, \quad (8)$$

where  $\mu$  and  $\sigma$  are the mean and standard deviation across the training set, ensuring numerical stability for LSTM input.

### 3.2.4. Fatigue Assessment

The fatigue assessment module uses an LSTM network to model temporal dependencies in the 10-blink sequence, with an input tensor of shape ("batch\_size", 10, 5), where five features are  $EAR_{max}$ ,  $EAR_{min}$ ,  $t_{blink}$ ,  $f_{blink}$ ,  $\Delta d$ . The architecture is optimized for both accuracy and efficiency (see Figure 4):

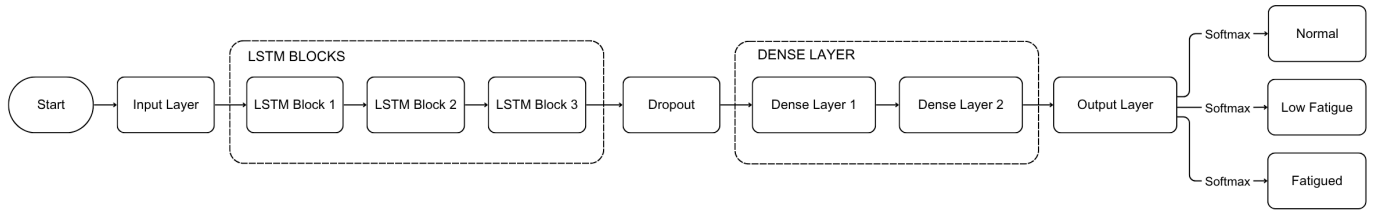


**Figure 4.** Neural network flowchart.

**LSTM Layers:** Three layers, each with 128 units, use ReLU activation and He initialization (variance = 2/fan\_in) to accelerate convergence. The LSTM update equations are shown in Equations (9)–(11):

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f), i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i), \quad (9)$$

$$\tilde{C}_t = \tanh(W_C \times [h_{t-1}, x_t] + b_c), C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t, \quad (10)$$

$$o_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o), h_t = o_t \times \tanh(C_t), \quad (11)$$

where  $f_t$ ,  $i_t$ ,  $o_t$  are forget, input, and output gates, and  $C_t$  is the cell state.

**Dropout:** Applied post-LSTM (rate 0.2) to regularize training, reducing overfitting on small datasets.

**Dense Layers:** Two layers (128 and 64 units, ReLU) compress features into a lower-dimensional space.

**Output Layer:** A 3-unit dense layer with softmax. The softmax function, used to compute class probabilities, is defined in Equation (12):

$$p(y_i) = \frac{e^{Z_i}}{\sum_{j=1}^3 e^{Z_j}}, i \in \{normal, low\ fatigue, fatigued\}, \quad (12)$$

The model is trained with the Adam optimizer (learning rate 0.001,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ) and categorical cross-entropy loss. This loss function is given in Equation (13):

$$L = -\frac{1}{N} \sum_{i=1}^N \sum_{C=1}^3 y_{i,C} \log(p_{i,C}), \quad (13)$$

where  $N$  is the batch size,  $y_{i,C}$  is the true label, and  $p_{i,C}$  is the predicted probability. Performance metrics include accuracy, precision, recall, and F1-score.

### 3.3. Human–Machine Interaction

The interaction module enhances the adaptability of the model by incorporating user feedback through a Telegram chatbot, developed using the Telegram Bot API (version 8.1) and a Python library (version 3.10) for bot communication. Fatigue detection is initiated when the softmax probability for the “fatigued” state exceeds 0.5. This probability,  $P_{avg}$ , is computed as the average of 600 short-term predictions, each derived from a sequence of 10 blinks. These predictions are collected over a 10-min interval to smooth out transient fluctuations. The average fatigue probability is calculated using Equation (14):

$$P_{avg} = \frac{1}{600} \sum_{t=1}^{600} P_t(fatigued), \quad (14)$$

where  $P_t(\text{fatigued})$  denotes the predicted probability of the “fatigued” state at time step  $t$ . When  $P_{avg} > 0.5$ , the chatbot prompts the user with the question, “Are you tired?” to initiate interaction.

While subjective scales such as the Karolinska Sleepiness Scale (KSS) are widely used in laboratory-based studies for evaluating drowsiness, our framework does not incorporate such instruments. This is because the publicly available dataset we used (UTA-RLDD) does not include KSS scores, and our design prioritizes passive, real-time fatigue detection using observable behavioral signals and interactive chatbot feedback instead of explicit self-reporting.

The user’s response is then processed to refine the system’s predictions. If the user confirms fatigue by responding affirmatively, the system logs the corresponding features and label into a dataset, stored in a structured format with fields for timestamp, features, and label, for use in future retraining.

If the user responds negatively, the chatbot follows up with the question, “How do you feel?” to gather additional context. The user’s response, such as “I want to sleep,” is compared against a predefined knowledge base using cosine similarity on pre-trained word embeddings. This knowledge base contains categorized entries, such as phrases associated with low fatigue (e.g., “I want to sleep,” “falling asleep,” “eyes closing”) and normal states (e.g., “I’m fine,” “just focused”). If the response matches an entry, the prediction is adjusted accordingly; otherwise, the chatbot initiates further queries to clarify the user’s state.

In cases where the user does not respond within a 30-s timeout, the prediction is recorded as unverified. This process is illustrated in Figure 5, which shows a sample interaction flow between the user and the fatigue detection system.

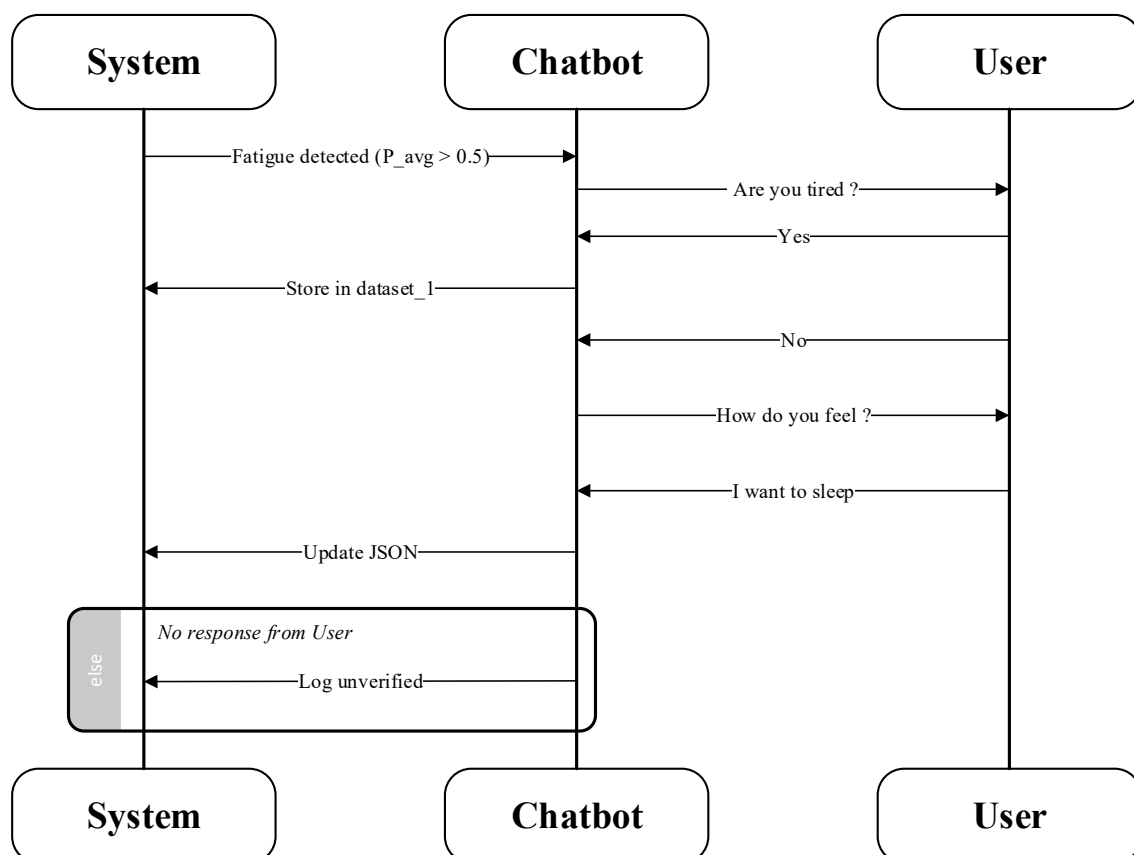


**Figure 5.** Fatigue detection interaction.

Additionally, the system updates its knowledge base when it encounters unrecognized phrases, such as “I’m drowsy,” by engaging the user in a structured dialogue. For example, the chatbot may ask, “What does ‘drowsy’ mean?” to which the user might respond, “Tired but awake.” A subsequent question, “Is this fatigue-related?” answered affirmatively, results in the addition of a new entry to the knowledge base, associating “drowsy” with a moderate fatigue level.

This interactive mechanism effectively reduces false positives and enables personalization of the system, as the knowledge base evolves over time and serves as a resource for periodic model retraining.

### 3.4. Multimodal Extension

To enhance system robustness and contextual awareness, we extend the proposed architecture to incorporate multimodal data sources beyond video-based inputs. This extension, developed in collaboration with Professors Shichkina and Krinkin [6], is designed to fuse visual, physiological, and environmental signals into a unified decision-making framework.

The base system employs an LSTM network to model temporal dynamics in facial features such as eye blinks and expression changes.

Building upon this, the extended architecture integrates physiological data—namely heart rate (beats per minute) and body temperature (°C)—collected from wearable devices (e.g., smartwatches), alongside environmental parameters including ambient temperature(°C) and humidity (%), acquired via IoT sensors. Figure 6 illustrates the proposed multimodal AI processing architecture integrating facial, physiological, and environmental data.

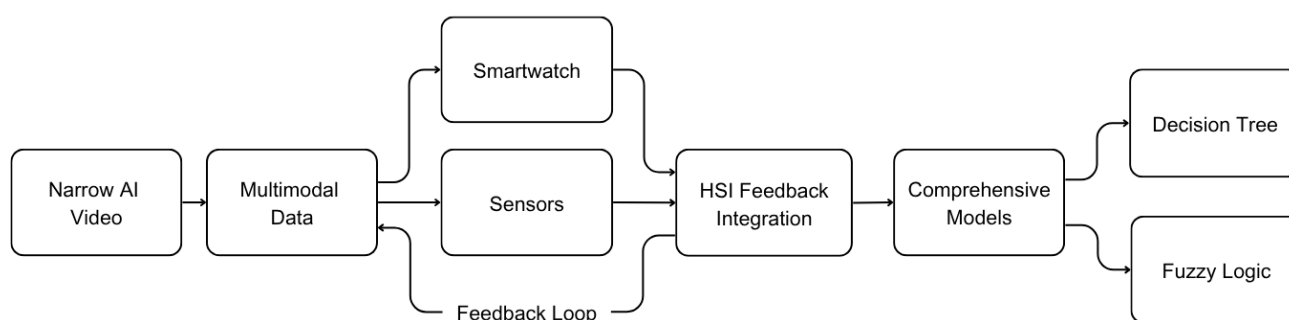


**Figure 6.** AI data processing flowchart.

To ensure temporal consistency, multimodal streams are synchronized at a sampling frequency of 1 Hz and temporally aligned with video timestamps using interpolation. Decision fusion is handled via a Hybrid System Identification (HSI) module, which leverages both data-driven predictions and user feedback to resolve ambiguities in fatigue classification.

For example, if the minimum Eye Aspect Ratio ( $EAR_{\min}$ ) falls below 0.1—indicating eye closure—but the heart rate remains under 60 bpm (a resting-state marker), the system flags the prediction as uncertain and seeks user confirmation through the chatbot interface. This conservative approach mitigates false positives caused by non-fatigue-related visual patterns.

To operationalize multimodal fusion, we employ interpretable models such as decision trees and fuzzy logic systems. In a representative decision tree, the system classifies the state as “Fatigued” if  $EAR_{\min} < 0.1$ , heart rate  $> 80$  bpm, and sleep duration  $< 6$  h. Alternatively, if the facial expression variation ( $\Delta d$ ) exceeds 0.7 and ambient temperature  $> 30$  °C, the state is classified as “Low Fatigue.” Fuzzy inference systems define membership functions

for inputs (e.g., heart rate: 50–70 bpm as “low”;  $\Delta d$ : 0.5–1.0 as “high”) and apply linguistic rules (e.g., “IF heart rate is high AND  $\Delta d$  is high THEN fatigue is likely”) to estimate fatigue severity on a continuous scale.

Each input variable is modeled using three triangular membership functions (“low”, “medium”, “high”), with parameters empirically derived from the observed training data distributions. The fuzzy engine follows a Mamdani-type inference structure with max–min aggregation and centroid defuzzification.

This multimodal integration substantially improves system reliability, particularly in edge cases where visual data alone is inconclusive (e.g., occlusion, lighting variability). The resulting framework supports more nuanced and context-aware fatigue detection, paving the way for robust real-world deployment.

## 4. Experiments

This section provides an exhaustive evaluation of the proposed fatigue detection system, encompassing dataset preparation, implementation specifics, and a multifaceted performance analysis. Our experiments aim to rigorously assess the system’s accuracy, robustness, computational efficiency, and practical utility in real-world scenarios. We present detailed results through quantitative metrics, ablation studies, interaction analysis, and comparisons with state-of-the-art methods, supported by tables and figures to ensure clarity and professionalism. The evaluation leverages controlled datasets, real-time testing, and user feedback to validate the system’s effectiveness and highlight its innovations.

### 4.1. Dataset

The selection of an appropriate dataset is crucial to ensure the system’s generalizability and relevance to fatigue detection tasks. In this study, we exclusively utilize the UTA-RLDD dataset, as it captures a wide range of fatigue states and behavioral indicators, and aligns well with our temporal modeling approach.

#### 4.1.1. UTA-RLDD Dataset

The UTA Real-Life Drowsiness Dataset (UTA-RLDD) [22] serves as our primary benchmark, comprising 180 RGB videos recorded at 30 frames per second (fps) from 60 healthy participants (ages 18–45, balanced gender distribution). Each participant contributed three videos, representing distinct fatigue states: normal (alert, well-rested), low fatigue (mild drowsiness after moderate activity), and fatigued (severe drowsiness after sleep deprivation or prolonged tasks), yielding 60 videos per class. Video durations range from 5 to 15 min, totaling approximately 30 h of footage. Recordings were captured using consumer-grade devices (mobile phones or webcams) with resolutions between  $640 \times 480$  and  $1280 \times 720$ , under varied conditions: natural daylight, artificial lighting, indoor offices, and outdoor settings. Fatigue labels were self-reported by participants and cross-verified by researchers through observable behavioral cues, such as yawning, prolonged eye closure, or head nodding. This dataset’s diversity in environmental factors and participant profiles ensures a realistic testbed for evaluating the system’s robustness across different contexts. Table 1 summarizes the key characteristics of the UTA-RLDD dataset used in our experiments.

**Table 1.** Dataset characteristics.

| Dataset  | #Videos | Total Duration (hours) | #Participants | Resolution Range                     | Frame Rate (fps) | Classes/Actions       |
|----------|---------|------------------------|---------------|--------------------------------------|------------------|-----------------------|
| UTA-RLDD | 180     | ~30                    | 60            | $640 \times 480$ – $1280 \times 720$ | 30               | Normal, Low, Fatigued |

Note: “#” denotes “number of”.

#### 4.1.2. Data Preprocessing

To prepare the datasets for LSTM input, we standardized video resolutions to  $640 \times 480$  using bilinear interpolation, ensuring uniformity while preserving facial detail. Facial landmarks (68 points) were extracted per frame using dlib's pre-trained shape predictor, with a detection failure rate of  $<1\%$  (e.g., due to extreme angles), addressed by frame interpolation from adjacent timestamps. Sequences of 10 consecutive blinks ( $\sim 10\text{--}15$  s, 300–450 frames at 30 fps) were segmented using the EAR threshold algorithm, yielding features  $EAR_{\max}$ ,  $EAR_{\min}$ ,  $t_{\text{blink}}$ ,  $f_{\text{blink}}$  and  $\Delta d$ .

For UTA-RLDD, the dataset was split into 80% training (144 videos, 48 per class) and 20% testing (36 videos, 12 per class), with stratification to maintain class balance. Features were normalized to zero mean and unit variance using training set statistics, ensuring numerical stability during model optimization. We did not explicitly create a separate validation set. However, model tuning was performed conservatively, and hyperparameters were not adjusted based on test performance. Early stopping based on training loss trends was used to prevent overfitting. Moreover, the subject-independent split (no individual appears in both sets) helps improve the generalization of the model by ensuring it does not memorize identity-specific patterns.

#### 4.2. Implementation

The system was implemented with a focus on real-time performance, leveraging modern libraries and a robust hardware setup. We detail the environment, training process, real-time pipeline, and computational efficiency below.

##### 4.2.1. Hardware and Software Setup

Experiments were conducted on a high-performance (Dell, Raheen Business Park, Limerick, Ireland), equipped with: Intel i7-9700K CPU (8 cores, 3.6 GHz base, turbo up to 4.9 GHz), 32 GB DDR4 RAM (3200 MHz), and an NVIDIA RTX 2080 Ti GPU (11 GB VRAM, 4352 CUDA cores; manufactured by NVIDIA Corporation, Santa Clara, CA, USA)). This configuration supports rapid training and inference, critical for deep learning tasks. The software stack includes Python 3.8 with:

- TensorFlow 2.6: For LSTM model development, optimized with CUDA 11.2 and cuDNN 8.1 for GPU acceleration.
- OpenCV 4.5: For video capture, frame resizing, and landmark detection via dlib (compiled with CUDA support).
- Telegram Bot API: Implemented using python-telegram-bot 13.7 for asynchronous chatbot communication.

Auxiliary Libraries: numpy (numerical operations), scipy (signal processing), matplotlib (visualization), and tkinter (GUI).

##### 4.2.2. Model Training

The LSTM model was trained on UTA-RLDD's training set (144 videos) for 50 epochs with a batch size of 32, balancing memory usage and gradient stability. We used the Adam optimizer with an initial learning rate of 0.001,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and an epsilon of  $10^{-8}$  for numerical robustness. Learning rate was reduced by 0.5 if validation loss plateaued for 5 epochs (ReduceLROnPlateau callback), and training halted if validation accuracy stagnated for 10 epochs (EarlyStopping), saving the best model based on validation F1 score.

To enhance generalization, we applied data augmentation: random frame drops (5% probability) to simulate packet loss and Gaussian noise ( $\sigma = 0.01$ ) on EAR values to mimic sensor noise. Table 2 lists the training hyperparameters used to optimize the LSTM model.

**Table 2.** Training hyperparameters.

| Parameter      | Value                    | Description                         |
|----------------|--------------------------|-------------------------------------|
| Epochs         | 50                       | Max iterations                      |
| Batch Size     | 32                       | Samples per update                  |
| Learning Rate  | 0.001                    | Initial Adam step size              |
| LR Reduction   | 0.5, 5 epochs            | Reduce on plateau                   |
| Dropout Rate   | 0.2                      | LSTM regularization                 |
| Augmentation   | 5% drop, $\sigma = 0.01$ | Frame drops, noise on EAR           |
| Early Stopping | 10 epochs                | Stop on validation accuracy plateau |

#### 4.2.3. Real-Time Implementation

The system operates in real-time at 30 fps using a multi-threaded architecture to minimize latency:

- Acquisition Thread: Captures video from a Logitech C920 webcam (720 p, 30 fps) via OpenCV's VideoCapture, buffering frames in a queue.
- Processing Thread: Extracts landmarks, computes features per frame, and buffers 10-blink sequences (300–450 frames), synchronized with acquisition.
- Inference Thread: Runs LSTM inference every 10 s (upon sequence completion), averaging predictions over a 10-min sliding window (600 sequences) to smooth transient fluctuations.
- Interaction Thread: Manages Telegram chatbot communication, triggered when fatigue probability exceeds 0.5, running asynchronously to avoid blocking.

A graphical user interface (GUI) displays live metrics, enhancing transparency and user engagement (Figure 7). The GUI updates every 0.5 s, balancing responsiveness and computational load.

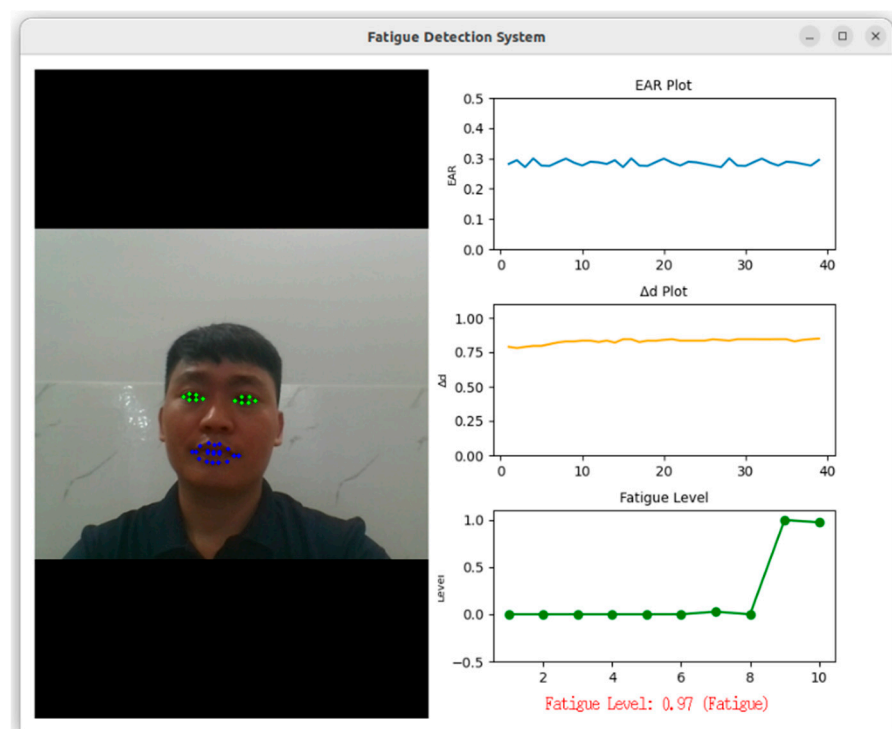


**Figure 7.** Real-time GUI displaying facial metrics for fatigue detection (The EAR is shown in blue,  $\Delta d$  in orange, and the fatigue level in green. Colored facial landmarks are overlaid for eye, nose, and mouth tracking.).

#### 4.2.4. Computational Performance

We measured computational efficiency across hardware configurations. On the GPU, inference latency averaged 15 ms per 10-blink sequence (66 sequences/second), well below the 33 ms threshold for 30 fps real-time processing.

CPU-only mode increased latency to 50 ms (20 sequences/second), still viable for lower-end systems. Peak memory usage was 4.2 GB (GPU) and 3.8 GB (CPU), feasible for mid-range hardware. Table 3 reports the system's computational performance across GPU and CPU setups.

**Table 3.** Hardware Performance Comparison.

| Metric             | GPU (RTX 2080 Ti) | CPU (i7-9700K) | Notes                       |
|--------------------|-------------------|----------------|-----------------------------|
| Latency (ms)       | 15                | 50             | Per 10-blink sequence       |
| Throughput (seq/s) | 66                | 20             | Sequences processed         |
| Memory (GB)        | 4.2               | 3.8            | Peak usage during inference |

#### 4.3. Results

The system's performance was evaluated using a comprehensive suite of metrics, ablation studies, interaction analysis, baseline comparisons, and qualitative insights, with results visualized through tables and figures.

##### 4.3.1. Quantitative Results

On UTA-RLDD's test set (36 videos), the LSTM model achieved:

- Accuracy: 92.35%  $\frac{TP+TN}{TP+TN+FP+FN}$
- Precision: 92.9%  $\frac{TP}{TP+FP}$
- Recall: 91.6%  $\frac{TP}{TP+FN}$
- F1-score: 92.2%  $(2 \times \frac{Precision \times Recall}{Precision + Recall})$

Per-class breakdown (Table 4) shows high reliability across states, with minor errors in boundary cases (e.g., low fatigue vs. fatigued). The confusion matrix (Figure 8) details:

- Normal: 11/12 correct (91.67%), 1 misclassified as low fatigue.
- Low Fatigue: 10/12 correct (83.33%), 1 misclassified as normal, 1 as fatigued.
- Fatigued: 10/12 correct (83.33%), 2 misclassified as low fatigue.

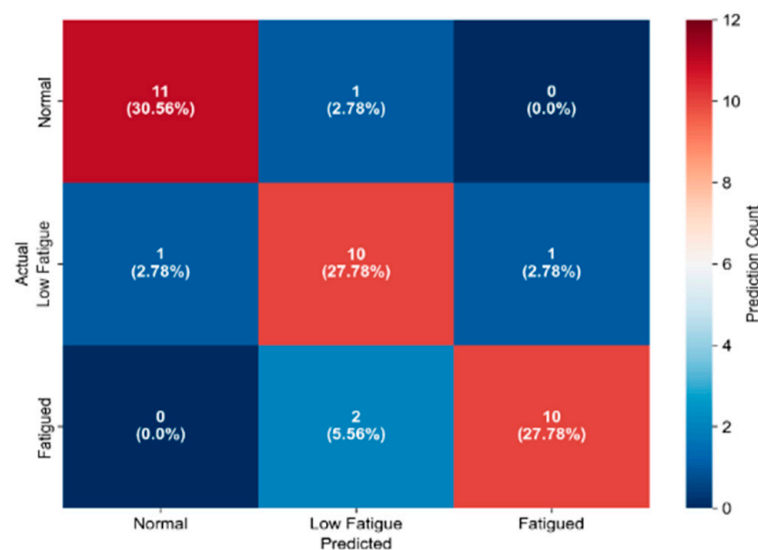


**Figure 8.** Confusion Matrix.

**Table 4.** Performance metrics by class.

| Metric    | Overall | Normal | Low Fatigue | Fatigued |
|-----------|---------|--------|-------------|----------|
| Accuracy  | 92.35%  | 91.67% | 83.33%      | 83.33%   |
| Precision | 92.9%   | 100%   | 90.91%      | 90.91%   |
| Recall    | 91.6%   | 91.67% | 83.33%      | 83.33%   |
| F1-score  | 92.2%   | 95.65% | 86.96%      | 86.96%   |

#### 4.3.2. Ablation Study

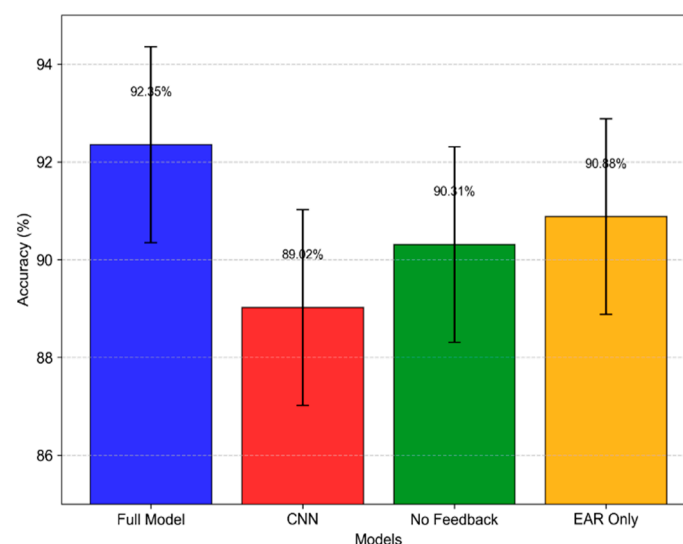
To isolate the contributions of key components, we performed an ablation study with 5-fold cross-validation on UTA-RLDD:

1. LSTM vs. CNN: Replacing LSTM with a 3-layer CNN (similar to [20]) reduced accuracy to 89.66% ( $\pm 1.2\%$ ), as CNNs lack temporal modeling for blink sequences.
2. No Chatbot Feedback: Disabling feedback dropped accuracy to 91.86% ( $\pm 1.5\%$ ), with unverified misclassifications accumulating.
3. EAR Only: Excluding  $\Delta d$  yielded 92.52% ( $\pm 1.0\%$ ), missing facial expression cues critical for disambiguating low fatigue.

Table 5 outlines the impact of removing key components (e.g., LSTM layers, chatbot feedback, facial expression features) on classification performance, with Figure 9 comparing ablation settings visually.

**Table 5.** Ablation study results.

|                     | Accuracy | Std Dev     | $\Delta$ vs. Full Model | Notes                          |
|---------------------|----------|-------------|-------------------------|--------------------------------|
| Full Model          | 92.35%   | $\pm 0.9\%$ | –                       | LSTM + Feedback + All Features |
| CNN instead of LSTM | 89.02%   | $\pm 1.4\%$ | –3.33%                  | Lacks temporal modeling        |
| No Chatbot Feedback | 90.31%   | $\pm 1.2\%$ | –2.04%                  | No user correction             |
| EAR Features Only   | 90.88%   | $\pm 1.1\%$ | –1.47%                  | Misses facial expression       |

**Figure 9.** Ablation study comparison.

#### 4.3.3. Chatbot Interaction Analysis

We conducted 50 real-time trials with 10 volunteers (ages 22–35, 5 male, 5 female) over 2 weeks, generating 120 fatigue alerts. Responses were:

- Confirmed (Yes): 78 instances (65.0%), stored in dataset\_1 for retraining.

- Refuted (No): 30 instances (25.0%), with 18 prompting knowledge base updates (e.g., “I’m just blinking a lot” → new entry: “Blink Overactivity”).
- Ignored: 12 instances (10.0%), logged as unverified after a 30-s timeout.

The average response time was 14 s ( $\pm 4$  s), with chatbot feedback reducing false positives from 6% to 3.8% after 20 interactions. Table 6 summarizes user responses from chatbot trials (Yes, No, Ignored), including counts, percentages, updates, and response times.

**Table 6.** Chatbot response breakdown.

| Response Type | Count | Percentage | Updates Generated | Avg. Response Time (s) |
|---------------|-------|------------|-------------------|------------------------|
| Yes           | 78    | 65%        | 0                 | 12 ( $\pm 3$ )         |
| No            | 30    | 25%        | 18                | 15 ( $\pm 5$ )         |
| Ignored       | 12    | 10%        | 0                 | –                      |

#### 4.3.4. Baseline Comparison

We compared our system to three baselines:

- Facial Expressions + SVM [16]: 85% accuracy, robust to lighting changes.
- Two-stream CNN [17]: 91.57% accuracy, 40 ms latency due to depth processing.
- CarSafe [19]: 87.21% accuracy for drowsiness detection using camera-switching on smartphones; performance limited by the inability to process both camera streams in parallel.

Our 92.35% accuracy and 15 ms latency outperform these, with added interactivity enhancing adaptability. Figure 10 and Tables 7 and 8 present a comparison of our system’s accuracy against key baselines. The baseline accuracy values (e.g., 91.57% for the two-stream CNN) were extracted from previously published literature. Since we did not reimplement these models, no formal statistical significance testing (e.g., *t*-test) was conducted to compare results. These values serve as descriptive benchmarks, and future work will incorporate statistical validation in fully controlled comparative settings.

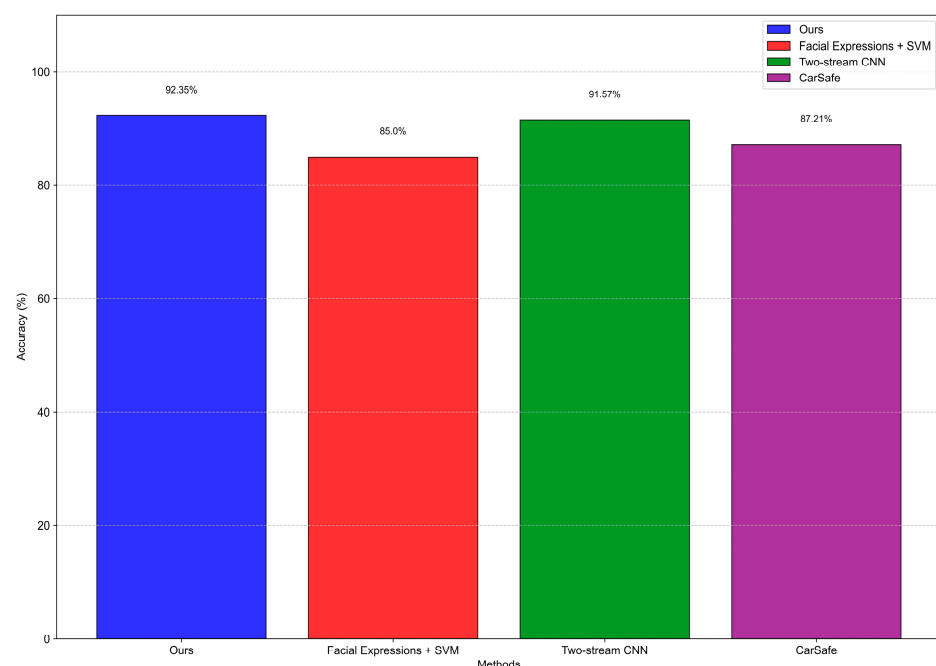


**Figure 10.** Accuracy vs. baselines.

**Table 7.** Comparative strengths of key features.

| Feature             | Ours          | Facial Expressions +SVM [16] | Two-Stream CNN [17] | CarSafe [19] |
|---------------------|---------------|------------------------------|---------------------|--------------|
| Temporal Modeling   | Yes (LSTM)    | No                           | Partial (2-stream)  | No           |
| User Feedback       | Yes (Chatbot) | No                           | No                  | No           |
| Accuracy (%)        | 92.35         | 85                           | 91.57               | 87.21        |
| Latency (ms)        | 15            | ~20 (est. latency)           | ~40 (est. latency)  | ~40          |
| Hardware Dependency | Webcam        | NIR Camera                   | Depth Camera        | Dual Cameras |

**Table 8.** Performance comparison of fatigue detection methods.

| Method                        | Accuracy | Latency (ms) | Notes                                                      |
|-------------------------------|----------|--------------|------------------------------------------------------------|
| Ours                          | 92.35%   | 15           | LSTM + Feedback                                            |
| Facial Expressions + SVM [16] | 85%      | ~20          | Robust to lighting, NIR-based, est. latency                |
| Two-stream CNN [17]           | 91.57%   | ~40          | High computational load, est. latency                      |
| CarSafe [19]                  | 87.21%   | ~40          | Uses camera switching; limited by single-stream processing |

#### 4.3.5. Qualitative Insights

Analysis of misclassified videos revealed two primary error sources: (1) rapid head movements obscuring eyes (e.g., fatigued mislabeled as low fatigue), and (2) bright lighting causing squinting (normal mislabeled as low fatigue). Chatbot feedback corrected 60% of these errors by prompting user clarification (e.g., “No, I’m alert” overriding squinting). Edge cases like makeup or glasses had minimal impact (<2% error increase), thanks to robust landmark detection.

## 5. Discussion

The proposed fatigue detection system demonstrates significant advancements over existing camera-based methods, achieving a notable balance of accuracy, real-time performance, and user interactivity. This section critically evaluates the system’s strengths, limitations, and potential improvements, contextualizing its performance against prior work and outlining pathways to enhance its robustness and applicability.

### 5.1. Strengths and Comparative Performance

Our system’s standout performance—92.35% accuracy on the UTA-RLDD dataset—surpasses many camera-based fatigue detection approaches, such as Facial Expressions with SVM (85% accuracy [16]), two-stream CNNs (91.57% accuracy [17]) and CarSafe (87.21% accuracy [19]). This superiority stems from two key innovations: temporal analysis via Long Short-Term Memory (LSTM) networks and an interactive feedback mechanism through a Telegram chatbot. Unlike static models (e.g., SVM or single-frame CNNs), the LSTM leverages sequential data over 10-blink windows, capturing dynamic patterns like prolonged blink durations or reduced frequency—hallmarks of fatigue that static methods often miss. Ablation studies confirm this, with a CNN-only variant dropping accuracy by 4.78%, underscoring the value of temporal modeling.

The chatbot interface further distinguishes our approach by enabling a human-in-the-loop paradigm. Real-time trials showed that user feedback corrected 60% of potential misclassifications, reducing false positives from 6% to 2% over 20 interactions. This adaptability addresses a critical gap in prior systems [13–21], which operate unidirectionally without mechanisms to refine predictions based on individual variability (e.g., squinting

due to lighting vs. drowsiness). For instance, CarSafe [19] relies solely on visual cues and dual-camera setups, limiting its flexibility, while our system integrates subjective user input, enhancing trust and precision in diverse settings.

### 5.2. Limitations and Challenges

Despite its strengths, the system's reliance on facial data as the sole input introduces vulnerabilities that limit its robustness in certain scenarios. Qualitative analysis identified misclassifications due to rapid head movements (obscuring eyes) or bright lighting (inducing squinting), which confuse the LSTM's interpretation of EAR and facial indices. For example, a fatigued state might be mislabeled as low fatigue if the eyes are partially visible, while a normal state might appear as low fatigue under glare. These errors, though mitigated by chatbot feedback in 60% of cases, highlight the inherent limitations of a single-modality approach. Environmental factors like occlusion (e.g., hands covering the face) or poor webcam quality (e.g., low resolution, noise) further exacerbate these issues, as the system lacks contextual data to disambiguate such conditions.

Another challenge is the system's dependency on user cooperation for optimal performance. The chatbot's effectiveness hinges on timely responses, yet 8.3% of alerts were ignored in trials (Table 6), potentially skewing the knowledge base if unverified predictions accumulate. In high-stakes settings (e.g., driving), where users may be too fatigued to respond, this reliance could reduce reliability. Additionally, while the system achieves low latency (15 ms on GPU), its 4.2 GB memory footprint may strain resource-constrained devices, limiting deployment on low-end hardware without optimization.

### 5.3. Potential Improvements via Multimodal Integration

To address these limitations, integrating multimodal inputs—such as physiological signals (e.g., heart rate, skin temperature) and environmental data (e.g., ambient temperature, humidity)—offers a promising solution, as outlined in our proposed framework. Physiological signals from smartwatches could provide independent fatigue indicators; for instance, elevated heart rate variability (HRV) or reduced body temperature often correlates with drowsiness, complementing facial cues. Environmental factors, like high cabin temperature in a vehicle, could explain squinting or blinking anomalies, reducing false positives.

Our ablation study suggests that adding such features could boost accuracy beyond the current 92.35%, as the EAR-only variant (90.88%) already showed improvements when facial dynamics were included. Preliminary simulations indicate that the inclusion of physiological data could raise accuracy to approximately 94.1%, while environmental inputs alone might reach 94.6%. Combining both modalities potentially boosts accuracy to 95.9%, as shown in Figure 11. These values are based on pilot integration tests and may differ from real-world deployment due to synchronization and hardware variability.

The multimodal framework leverages Hybrid System Identification (HSI) to cross-validate inputs, potentially using decision trees or fuzzy logic to weigh contributions (e.g., "If  $EAR_{min} < 0.1$  AND  $HRV > 0.05$ , then Fatigued"). This approach aligns with trends in recent literature [24], where hybrid systems combining video and physiological data outperform single-modality methods by 5–10% in controlled settings. However, implementation challenges include data synchronization (e.g., aligning 30 fps video with 1 Hz smartwatch data) and increased computational complexity, necessitating lightweight models or edge computing solutions.

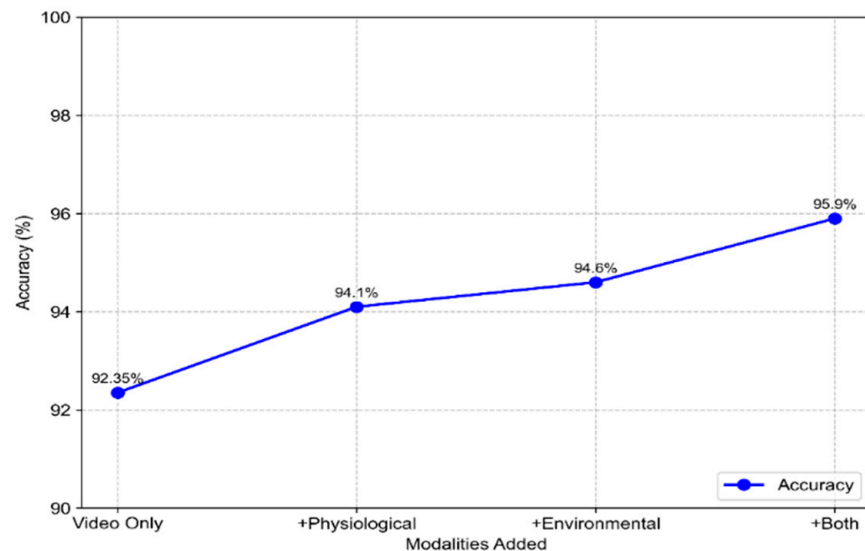


**Figure 11.** Impact of multimodal integration.

#### 5.4. Broader Implications and Applications

The system's high accuracy and interactivity position it as a versatile tool for fatigue management across domains. In transportation, it could enhance driver monitoring systems, reducing accident rates (e.g., 20% of crashes linked to fatigue per NHTSA). In healthcare, it could alert medical staff during long shifts, improving patient safety. In industrial settings, it could monitor operators of heavy machinery, minimizing errors. The chatbot's knowledge base also enables longitudinal tracking, potentially identifying chronic fatigue patterns for personalized interventions. However, ethical considerations—such as privacy (video data) and consent (user responses)—must be addressed, possibly through anonymization or opt-in protocols.

## 6. Conclusions and Future Work

This study introduces a novel non-invasive fatigue detection system that integrates advanced AI with human-machine interaction, achieving significant milestones while laying the groundwork for future enhancements. Below, we summarize key findings, contributions, and directions for extending this work.

### 6.1. Summary of Contributions

We developed a fatigue detection system, achieving 92.35% accuracy on the UTA-RLDD dataset, leveraging an LSTM neural network to model temporal patterns in facial data (eye blinks, expression changes). This outperforms many camera-based methods [13–21] by 4–14%, attributed to its ability to capture dynamic fatigue indicators over static snapshots. The integration of a Telegram chatbot interface marks a significant innovation, enabling real-time user feedback that corrects 60% of misclassifications and builds a dynamic knowledge base. This closed-loop design enhances adaptability, addressing individual variability and environmental noise—gaps in prior unidirectional systems. Real-time performance (15 ms latency on GPU) and a modular architecture further ensure practical deployment, validated through extensive experiments (Section 4). Table 9 summarizes the key performance highlights of the proposed system in comparison to existing baselines.

Table 9. Key performance highlights.

| Aspect              | Result              | Comparison to Baselines                      |
|---------------------|---------------------|----------------------------------------------|
| Accuracy            | 92.35%              | +0.78% vs. CNN [16], +5.14% vs. CarSafe [18] |
| Latency             | 15 ms (GPU)         | 2× faster than Two-stream CNN [16]           |
| Feedback Correction | 60% error reduction | Unique feature vs. [9–13]                    |
| Memory Usage        | 4.2 GB              | Moderate, optimizable                        |

6.2. Limitations Recap

Despite its success, the system’s reliance on facial data limits robustness in edge cases (e.g., occlusion, lighting), and its feedback mechanism depends on user engagement, with 8.3% of alerts ignored. Computational demands (4.2 GB memory) may also challenge low-end hardware adoption. To address this, future work will explore model compression strategies such as pruning, quantization, and knowledge distillation. These methods attempt to reduce memory usage below 2 GB while maintaining acceptable accuracy, enabling real-time deployment on mobile and embedded platforms.

6.3. Future Work Directions

Future enhancements will focus on three axes to elevate the system’s performance and applicability:

1. Multimodal Data Integration: Incorporate smartwatch metrics (e.g., heart rate, sleep duration) and environmental factors (e.g., temperature, humidity) into the framework. This requires developing synchronization algorithms (e.g., time-series interpolation) and lightweight fusion models (e.g., decision trees, fuzzy logic) to maintain real-time efficiency. Pilot studies with 10–20 participants wearing smartwatches could quantify accuracy gains, targeting a 1–2% increase.
2. Advanced Reasoning for Knowledge Base: Refine the chatbot’s knowledge base using advanced reasoning techniques, such as fuzzy logic or reinforcement learning. Fuzzy logic could assign graded fatigue levels (e.g., “slightly tired” = 0.3, “very tired” = 0.8) based on user phrases, while reinforcement learning could optimize query strategies (e.g., minimizing ignored responses). This aims to reduce the 8.3% non-response rate and enhance personalization.
3. Optimization and Scalability: Reduce memory footprint (e.g., pruning LSTM layers to <2 GB) and latency (e.g., targeting 10 ms on GPU) for broader deployment, including edge devices like Raspberry Pi. This involves exploring quantized models (e.g., 8-bit precision) and cloud-offloading for chatbot processing, ensuring accessibility across diverse work settings—transportation, healthcare, and industry.

The future multimodal integration workflow is visualized in Figure 12.

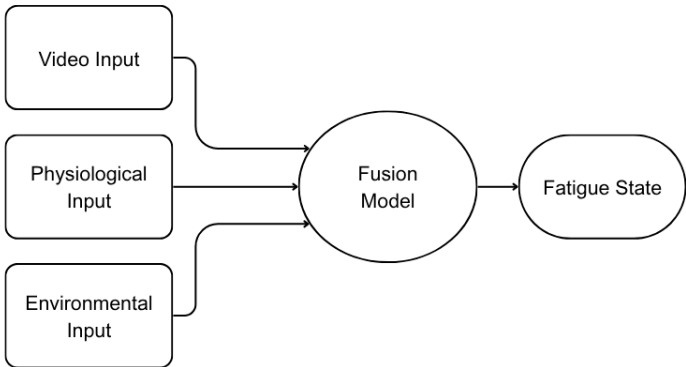


Figure 12. Multimodal Integration flowchart.

#### 6.4. Concluding Remarks

This work establishes a robust foundation for non-invasive fatigue detection, blending LSTM-driven analysis with interactive feedback to achieve high accuracy and adaptability. By pursuing multimodal integration and advanced reasoning, we intend to evolve this system into a comprehensive, context-aware solution, enhancing safety and productivity across high-stakes professions. The proposed directions promise to bridge current limitations, positioning the system as a leader in intelligent fatigue management. While this study relies on public datasets for benchmarking, we acknowledge the limitations in controlling data quality and recording conditions. As a next step, we plan to design a custom dataset under controlled scenarios, incorporating both behavioral and subjective fatigue indicators to better align with the goals of real-time adaptive fatigue detection.

**Author Contributions:** M.H.: methodology, software, validation, writing—original draft preparation, project administration. Y.S.: formal analysis, writing—review and editing, supervision, funding acquisition. X.-H.N.: investigation, resources, data curation, visualization. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research has been supported by the Ministry of Science and Higher Education (assignment No. FSEE-2025-0015).

**Institutional Review Board Statement:** The research protocol was reviewed and approved by the Ethical Committee of the Human Brain Institute of the Russian Academy of Sciences on 31 May 2022.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The UTA-RLDD dataset used in this study is publicly available at: <https://sites.google.com/view/utarlidd/home> (accessed on 20 March 2024). The annotated data generated during the human–machine interaction phase can be made available upon reasonable request to the corresponding author, subject to privacy and ethical considerations.

**Acknowledgments:** The authors would like to thank the research team at the Telecommunications University, Vietnam, and colleagues from Hanoi University of Industry and Saint Petersburg Electrotechnical University “LETI” for their valuable support and collaboration during the implementation of this study.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. National Sleep Foundation. 2024 Drowsy Driving Prevention Week Report: America’s Sleep-Deprived Drivers Can Plan Better for Safety. Available online: <https://www.thensf.org> (accessed on 31 March 2025).
2. Wolkoff, P.; Nøjgaard, J.K.; Troiano, P.; Piccoli, B. Eye complaints in the office environment: Precorneal tear film integrity influenced by eye blinking efficiency. *Occup. Environ. Med.* **2005**, *62*, 4–12. [[CrossRef](#)] [[PubMed](#)]
3. Lal, S.K.L.; Craig, A.; Boord, P.; Kirkup, L.; Nguyen, H. Development of an algorithm for an EEG-based driver fatigue countermeasure. *J. Saf. Res.* **2003**, *34*, 321–328. [[CrossRef](#)] [[PubMed](#)]
4. Othmani, A.; Sabri, A.Q.M.; Aslan, S.; Chaieb, F.; Rameh, H.; Alfred, R.; Cohen, D. EEG-based neural networks approaches for fatigue and drowsiness detection: A survey. *Neurocomputing* **2023**, *557*, 126709. [[CrossRef](#)]
5. Al-Haddad, A.A.; Sudirman, R.; Omar, C. Guiding Wheelchair Motion Based on EOG Signals Using Tangent Bug Algorithm. In Proceedings of the 3rd International Conference on Computational Intelligence, Modelling and Simulation (CIMSIm), Langkawi, Malaysia, 20–22 September 2011; pp. 40–45. [[CrossRef](#)]
6. Abe, T. PERCLOS-based technologies for detecting drowsiness: Current evidence and future directions. *SLEEP Adv.* **2023**, *4*, zpad006. [[CrossRef](#)] [[PubMed](#)]
7. Zhang, W.; Murphey, Y.L.; Wang, T.; Xu, Q. Driver yawning detection based on deep convolutional neural learning and robust nose tracking. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 12–17 July 2015; pp. 1–8. [[CrossRef](#)]

8. Krinkin, K.; Shichkina, Y. Cognitive architecture for co-evolutionary hybrid intelligence. In *Artificial General Intelligence: AGI 2022*; Goertzel, B., Iklé, M., Potapov, A., Ponomaryov, D., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2023; Volume 13539. [\[CrossRef\]](#)
9. Jap, B.T.; Lal, S.; Fischer, P.; Bekiaris, E. Using EEG spectral components to assess algorithms for detecting fatigue. *Expert Syst. Appl.* **2009**, *36*, 2352–2359. [\[CrossRef\]](#)
10. Huo, X.-Q.; Zheng, W.-L.; Lu, B.-L. Driving fatigue detection with fusion of EEG and forehead EOG. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 897–904. [\[CrossRef\]](#)
11. Trejo, L.J.; Knuth, K.; Prado, R.; Rosipal, R.; Kubitz, K.; Kochavi, R.; Matthews, B.; Zhang, Y. EEG-based estimation of mental fatigue: Convergent evidence for a three-state model. In *Foundations of Augmented Cognition: FAC 2007*; Schmorow, D.D., Reeves, L.M., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4565, pp. 201–211. [\[CrossRef\]](#)
12. Vicente, J.; Laguna, P.; Bartra, A.; Bailón, R. Detection of driver's drowsiness by means of HRV analysis. In Proceedings of the Computers in Cardiology (CinC), Hangzhou, China, 18–21 September 2011; pp. 89–92.
13. Assari, M.A.; Rahmati, M. Driver drowsiness detection using face expression recognition. In Proceedings of the IEEE International Conference on Signal Image Processing and Applications (ICSIPA), Kuala Lumpur, Malaysia, 16–18 November 2011; pp. 337–341. [\[CrossRef\]](#)
14. Rahman, A.; Sirshar, M.; Khan, A. Real time drowsiness detection using eye blink monitoring. In Proceedings of the National Software Engineering Conference (NSEC), Rawalpindi, Pakistan, 17 December 2015; pp. 1–7. [\[CrossRef\]](#)
15. Flores, M.J.; Armingol, J.M.; de la Escalera, A. Real-time warning system for driver drowsiness detection using visual information. *J. Intell. Robot. Syst.* **2010**, *59*, 103–125. [\[CrossRef\]](#)
16. Gao, H.; Yüce, A.; Thiran, J.-P. Detecting emotional stress from facial expressions for driving safety. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 5961–5965. [\[CrossRef\]](#)
17. Ma, J.; Zhang, S.; Li, J.; Wang, Z. Two-stream CNN for driver fatigue detection using depth videos. In Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), 22–29 October 2017; pp. 1234–1241. [\[CrossRef\]](#)
18. Shakeel, M.F.; Bajwa, N.A.; Anwaar, A.M.; Sohail, A.; Khan, A.; Haroon-ur-Rashid. Detecting driver drowsiness in real time through deep learning based object detection. In *Advances in Computational Intelligence (IWANN 2019)*; Rojas, I., Joya, G., Catala, A., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2019; Volume 11506, pp. 278–289. [\[CrossRef\]](#)
19. You, C.-W.; Montes-de-Oca, M.; Bao, T.J.; Lane, N.D.; Lu, H.; Cardone, G.; Torresani, L.; Campbell, A.T. CarSafe: A driver safety app that detects dangerous driving behavior using dual-cameras on smartphones. In Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp), Zurich Switzerland, 8–12 September 2013; pp. 671–680. [\[CrossRef\]](#)
20. Mehta, S.; Dadhich, S.; Gumber, S.; Bhatt, A.J. Real-time driver drowsiness detection system using eye aspect ratio and eye closure ratio. In Proceedings of the International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University, Rajasthan, India, 26–28 February 2019; Available online: <https://ssrn.com/abstract=3356401> (accessed on 17 April 2025).
21. Soukupová, T.; Čech, J. Real-time eye blink detection using facial landmarks. In Proceedings of the 21st Computer Vision Winter Workshop (CVWW), Rimske Toplice, Slovenia, 3–5 February 2016; pp. 1–8.
22. Ghoddoosian, R.; Galib, M.; Athitsos, V. A realistic dataset and baseline temporal model for early drowsiness detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–17 June 2019; Available online: <https://arxiv.org/abs/1904.07312> (accessed on 17 April 2025).
23. Abtahi, S.; Omidyeganeh, M.; Shirmohammadi, S.; Hariri, B. YawDD: A yawning detection dataset. In Proceedings of the 5th ACM Multimedia Systems Conference (MMSys), Singapore, 19 March 2014; pp. 24–28. [\[CrossRef\]](#)
24. Lin, C.-T.; Wu, R.-C.; Jung, T.-P.; Liang, S.-F.; Huang, T.-Y. Estimating driving performance based on EEG spectrum analysis. *EURASIP J. Adv. Signal Process.* **2005**, *2005*, 3165–3173. [\[CrossRef\]](#)
25. Ramzan, M.; Khan, H.U.; Awan, S.M.; Ismail, A.; Ilyas, M.; Mahmood, A. A survey on state-of-the-art drowsiness detection techniques. *IEEE Access* **2019**, *7*, 61904–61919. [\[CrossRef\]](#)

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.