Received: October 20, 2024. Revised: December 9, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Drowsiness Classification Using ResNet50 and Time Series Transformer Based on Blink Pattern Features

Ahmad Zaini[1] I Ketut Eddy Purnama[2] Yoyon Kusnendar Suprapto[2]
Eko Mulyanto Yuniarno[2]*

[1]*Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Indonesia*
[2]*Department of Computer Engineering, Institut Teknologi Sepuluh Nopember, Indonesia*
* Ekomulyanto@ee.its.ac.id

Abstract: Drowsiness classification faces major technical challenges in accurately capturing long-term temporal patterns while coping with disturbances such as inconsistent lighting and head motion. Traditional approaches based on Eye Aspect Ratio (EAR) analysis or facial landmarks are often susceptible to environmental noise and require complex data preprocessing, which reduces their reliability in real-world conditions. This study proposes a deep learning-based framework that combines ResNet50 and Time Series Transformer (TST) to improve the performance of drowsiness classification. ResNet50 is used to detect eye conditions and generate binary blink patterns, and TST captures the temporal dependencies in these blink patterns. By extracting statistical features such as mean, variance, and blink duration, this approach simplifies the data preprocessing process and improves the model's robustness to environmental disturbances. The experimental results demonstrate that the proposed framework achieves 97% accuracy with comparable precision, recall, and F1 score, outperforming conventional methods in modeling temporal patterns and dealing with technical disturbances. The proposed method exhibits high computational efficiency and provides a practical and reliable solution for real-time drowsiness classification.

Keywords: Drowsiness classification, Blink pattern analysis, ResNet50, Time series transformer.

1. Introduction

The WHO's Global Status Report on Road Safety 2023 states that 1.19 million deaths occurred due to road traffic accidents in 2021, corresponding to a rate of 15 deaths per 100,000 people [1]. Motorcyclists and private car drivers account for approximately 25% of the most severe accidents, with driver negligence being a primary cause. Often, drivers lose concentration or experience fatigue, leading to microsleep, a brief period of unconsciousness lasting up to 15 seconds [2]. Microsleep occurs when individuals struggle to stay awake, often without realizing it [3, 4]. In driving conditions, microsleep can lead to accidents, resulting in fatalities. This underscores the importance of research into early drowsiness detection to provide timely warnings to drivers.

Numerous studies have been conducted on early driver warning systems for drowsiness, employing both invasive and non-invasive techniques. In general, there are three main approaches to detecting drowsiness: monitoring steering patterns during driving [5], assessing physiological conditions [6], and analyzing driver behavior [7].

First, the driving steering pattern is observed by monitoring the driver's actions while driving [8]. Several sensors are installed on the vehicle, including those for speed, acceleration, engine speed, and steering speed, as well as other sensors connected to the Electronic Control Unit (ECU). The sensor data are processed to classify the driving style and assess the consistency of driving patterns. Consistency in speed, acceleration, and steering rotation is used to determine whether the driver's concentration is impaired or reduced [8]. Second, the physical and psychological condition of the driver is monitored.

This method involves measuring biological signals from the driver, typically using EEG (Electroencephalography) sensors [3-5, 9, 10], ECG (Electrocardiography) [11], EOG (Electrooculography) [9, 12], and EMG (Electromyography) [13, 14]. However, both of these methods have limitations in application due to the potential for significant signal noise caused by the driver's movements. Additionally, such techniques may affect driver comfort.

The third approach is a non-invasive method that relies on visual data collected via camera sensors. This method extracts facial features, such as the eyes, nose, mouth, and facial expressions, from the facial image data. Eye features are used to monitor blink patterns [15, 16], while mouth features capture yawning patterns. Overall facial features are analyzed to detect changes in expression [17].

Several non-invasive approaches are employed for facial image processing to extract these facial features. Image classification, object detection, and segmentation techniques often use deep learning models based on Convolutional Neural Network (CNN). In addition to traditional statistical methods such as Principal Component Analysis (PCA) and Support Vector Machines (SVM), deep learning models have been explored for drowsiness detection. For instance, S. Park et al. [10] compared different architectures for alarm detection applications. They utilized three networks: the AlexNet architecture, which includes five CNN layers and three Fully Connected (FC) layers [11], for face detection; the Visual Geometry Group (VGG)-FaceNet architecture with 16 layers for facial feature extraction; and the FlowImageNet network for behavioral analysis in detecting drowsiness.

Although various approaches have been proposed, such as driving pattern analysis, physiological signals, and facial landmark pattern processing for detecting drowsiness [18], major challenges remain in capturing long-term temporal patterns and overcoming disturbances such as inconsistent lighting and head motion. This study proposes a framework that combines ResNet50 and TST models to overcome the limitations of previous drowsiness detection methods. ResNet50 is used to accurately classify eye conditions from Near-Infrared (NIR) images and generate binary blink patterns, while TST is used to capture the temporal dependencies in these blink patterns. The Time Series Transformer (TST) is a neural network architecture designed to process and analyze sequential data, specifically time series. This combination ensures robustness to noise and environmental variations, such as changing lighting and head motion. By generating binary blink patterns and extracting their statistical features, such as the distribution and duration of eye opening and closing, this framework simplifies the data preprocessing process without compromising classification performance. The contributions of this study are as follows:

1. The proposed framework uses the statistical features of the duration and distribution of eye blinks and uses ResNet50 to generate binary blink patterns and a Time Series Transformer (TST) to classify sleepiness into three categories.

2. This approach minimizes reliance on the accuracy of facial and eye landmarks, which has been a focus in prior studies using EAR spatial patterns [19, 20].

3. The use of low-resolution 8-bit NIR video data (512 × 424) for training and testing presents challenges for TST to achieve accurate classification results.

The rest of this paper is organized as follows: Section 2 presents a related work review on drowsiness detection. Section 3 describes the methodology, including data preprocessing, feature extraction, and model architecture. Section 4 presents the experimental results and compares the proposed framework to existing methods. Finally, Section 5 presents the conclusions of this study and directions for future research.

2. Related work

Non-invasive detection methods measure the Eye Aspect Ratio (EAR) and the Percentage of Eyelid Closure (PERCLOS) based on visible images. The time the eyes remain closed is measured as a percentage over a specific period, with P70, P80, and EYEMEAS (EM) being the three primary measurement methods. P80 is regarded as the most reliable indicator of fatigue [21], outperforming other measures, including general Eye-Tracking Signal (ETS) [13]. This method requires accurate detection of eye opening and closing, as well as measurement of the duration of eye closure. The challenge lies in ensuring the accuracy of eye detection in both open and closed states.

Yang et al. [17] proposed a Video-Based Driver Drowsiness Detection (VBDDD) with Optimised Utilisation of Key Facial Features (VBFLLFA) method that exploits facial landmarks and local facial areas (eyes and mouth) to detect drowsiness. They uses video VBDDD, YawDD, and NTHU-DDD dataset. This method uses the Common Spatial Pattern (CSP) algorithm for spatial filtering, enhancing inter-class discrimination, and the Two-

Branch Multi-Head Attention (TB-MHA) module for spatial and temporal feature extraction. The center loss with center vector distance penalty is also applied to improve class separation in the feature space. The advantages of this method include the optimization of spatial-temporal features and reduction of video data redundancy; however, it relies on the accuracy of facial landmark detection and requires high computing power, limiting real-time application.

Bai et al. [22] introduced a novel approach for driver drowsiness detection using a two-stream Spatial-Temporal Graph Convolution Network (2s-STGCN). They uses video VBDDD, YawDD, and NTHU-DDD dataset. This method effectively combines spatial and temporal features from driver facial videos, addressing challenges such as variations in lighting, obstructions, shadows, and head pose. Additionally, Han et al. [23] proposed a multimodal fatigue recognition system that uses a Temporal Convolutional Network (TCN) to process EAR sequences and ResNet3D to process eyelid image sequences. This combination enables effective spatial and temporal analysis, thereby improving fatigue detection accuracy. An annealing-based learning rate decay algorithm was applied to prevent the model from getting stuck in a local solution during training. They utilizes the University of Texas at Arlington Real-Life Drowsiness Dataset (UTA-RLDD), and the ULg Multimodality Drowsiness Database (DROZY). However, the method discussed in this paper relies heavily on consistent lighting and precise facial and eye landmark detection. Methods like those in [18-20], which use graphical construction as a basis for analysis, are prone to failure when landmark detection is inaccurate or absent. Furthermore, blink detection based on a threshold EAR value is not universally applicable.

Near-infrared imaging (NIR) is a method that utilizes near-infrared light, which is beyond the range of human vision, to examine the various properties of objects or systems [24]. NIR imaging can vary in spatial resolution and wavelength depending on the objective and technique. Small particles scatter incident light, altering the intensity of the light at specific wavelengths. When the particle size is very small (< λ/10), light follows Rayleigh scattering, described by the equation:

$$L_s = \frac{L_0}{\lambda^4} \tag{1}$$

Where L_s is the intensity of scattered light, and L_0 is the intensity of the incident light. Since the NIR wavelength band (700 – 1100 nm) is longer than the visible band (400 – 700 nm), Eq. (1) indicates that NIR images experience less scattering compared to visible images.

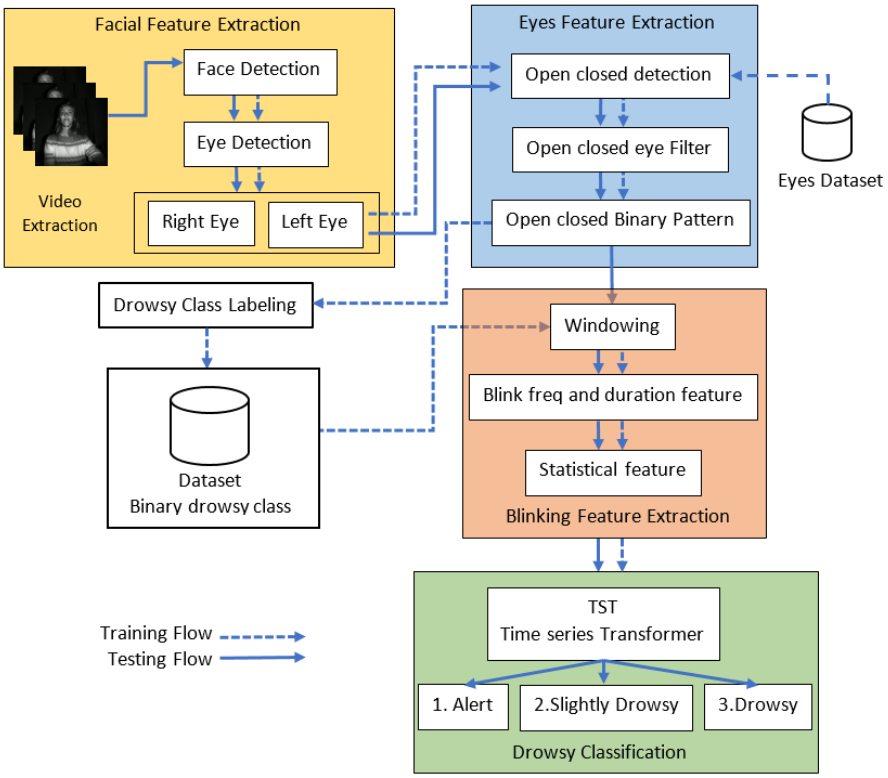

Figure. 1 BPF Framework Proposed Method

3. Methodology

In this section, we present a specialized technique designed to classify participant drowsiness into three classes, as shown in Table 1, using a framework called Blinking Pattern Feature (BPF). The BPF method utilizes video signals as visual data input and combines ResNet50 with a Time Series Transformer (TST) to classify drowsiness levels. The entire process is divided into facial feature extraction, eye feature extraction, blink pattern extraction, and drowsiness classification. The BPF framework for drowsiness classification is illustrated in Fig. 1.

3.1 Dataset

This study was obtained from the Open Repository and Bibliography (ORBi) "ULg Multimodality Drowsiness Database" or Drozy[25]. This dataset includes ECG, EOG, EMG data, and 8 bit NIR video recordings from 14 participants under three conditions, classified according to [26]. We used 8-bit video NIR with a resolution of 512 x 424 30 fps, with modifications to align with classification classes based on the Karolinska Sleepiness Scale (KSS) [25].

The system distinguishes between "Alert", "Slightly Drowsy", and "Drowsy" states based on the established Karolinska Sleepiness Scale, a widely recognized tool for assessing drowsiness. In this classification scheme, individuals scoring between 1 and 3 on the Karolinska scale are categorized as "Alert," representing states of wakefulness and attentiveness according to the scale's guidelines.

Table 1. Relationship between the Karolinska Sleepiness Scale and drowsiness classification.

Scale	Karolinska Sleepiness Scale	Class
1	Extremely Alert	1. Alert
2	Very Alert	
3	Alert	
4	Fairly Alert	2. Slightly Drowsy
5	Neither Alert nor Sleepy	
6	Some signs of sleepiness	
7	Sleepy, but no effort to stay alert	3. Drowsy
8	Sleepy, some effort to stay alert	
9	Very Sleepy, great effort to stay alert	

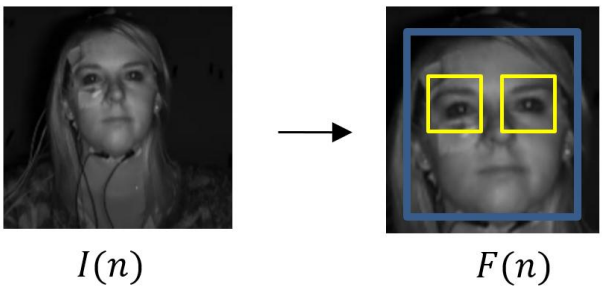

Figure. 2 Multi-frame Face Array

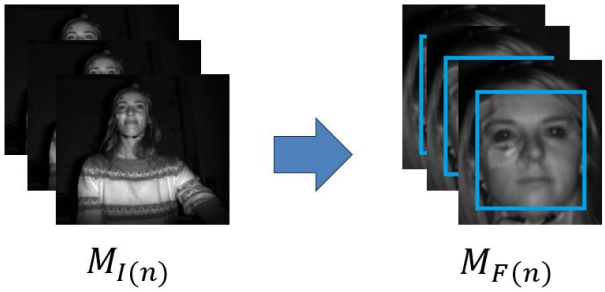

Figure. 3 Facial Feature Detection

The second classification scheme applies to individuals who scored between 4 and 6 on the Karolinska scale and were categorized as "Slightly Drowsy". This range typically represents varying levels of drowsiness or reduced alertness, as defined by the Karolinska scale. The third classification includes those who scored between 7 and 9 on the Karolinska scale and were classified as "Drowsy". These scores generally indicate that the participant is experiencing difficulty maintaining alertness.

All classifications 1 to 3 for "Alert," 4 to 6 for "Slightly Drowsy", and 7 to 9 for "Drowsy" are explicitly detailed in Table 1 and serve as the foundational criteria for the categorization process. The proposed method provides a standardized, evidence-based approach for distinguishing between different states of alertness and drowsiness.

3.2 Facial feature extraction

Facial feature extraction is the process of identifying and extracting key information from a face. The extracted facial features usually include distinct geometric and textural characteristics, such as the distance between the eyes, nose length, jaw shape, and lip structure. In this study, facial feature extraction focuses on isolating the eyes as the Region of Interest (ROI), which is the primary feature used for drowsiness classification.

The video dataset is first extracted as a multiframe image $M_{I(n)}$. For **face detection**, the goal is to isolate the face area as the ROI.

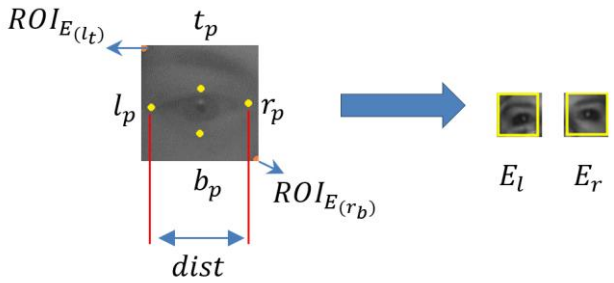

Figure. 4 Eyes Region Of Interest

This stage involves identifying facial features in a single image frame, as shown in Fig. 2. Face detection for each image *I(n)* was performed using the OpenCV library and the method proposed by Viola and Jones [27]. They used the Haar feature-based cascade classifier, an effective object detection method, to identify the face ROI $F_{(n)}$. As a result, the full face detection data from the video can be represented as a matrix $M_{F(n)}$, shown in Fig. 3. After face detection, eye features are then identified.

Eye detection: The goal of eye detection is to obtain the ROI for the right and left eyes, as shown in Fig. 4. The object detection method implemented in this study combines a Support Vector Machine (SVM) and Histogram of Oriented Gradients (HOG). HOG and SVM are widely used for object detection in the medical field, in addition to face detection and local facial feature detection [28]. The facial feature detection technique from the Dlib library originated from a method introduced in the paper "Histograms of Oriented Gradients for Human Detection" [29]. In this study, HOG was used as a feature descriptor for object detection, and it has been widely adopted for detecting faces and facial features [30]. The Dlib library was used to obtain four landmark points for each eye (left point l_p, right point r_p, top point t_p, and bottom point b_p) for both the right and left eyes. These points were then used to define the ROI for the eyes $ROI_{E(lt,rb)}$ (Eqs. (5) to (6)):

$$x_{cent},\ y_{cent} = l_p + \frac{r_p - l_p}{2}, t_p + \frac{b_p - t_p}{2} \quad (2)$$

$$dist = r_p - l_p \quad (3)$$

$$ofs = \left(1.7 \times \frac{dist}{2}\right) \quad (4)$$

$$ROI_{E(l_t)} = (x_{cent} - ofs, y_{cent} - ofs) \quad (5)$$

$$ROI_{E(r_b)} = (x_{cent} + ofs, y_{cent} + ofs) \quad (6)$$

dist is the distance between the right and left points of the eye landmark. *ofs* is the offset parameter used to determine the coordinates of the upper left eye and lower right eye ROI coordinates. The NIR video dataset was extracted into multiple frames, producing approximately 7,000 to 18,000 frames for a 10-minute video at 30fps. At this stage, two ROIs for the right eye E_r and left eye E_l are obtained from each image frame where facial features are successfully detected.

3.3 Eye feature extraction

The eye feature extraction stage is designed to obtain the eye blink pattern in binary format over a specific period *T* from each input video. The binary format uses '1' and '0' to represent when the eyes are detected as open or closed, respectively, in each frame of the facial feature *F(n)*. The eye-blink pattern in binary format is represented as a binary sequence such as "10001111001," which corresponds to eye blinks over a certain period. This allows for the calculation of both the blink frequency and the duration of each blink. The substages within this process include the following:

Open/Closed Classification: This classification identifies eye blink patterns by determining whether the eyes are open or closed. The classification is based on the ResNet50 architecture model. ResNet50 is a deep learning model with 50 layers, built using the concept of residual learning (Fig. 5)[31], which allows for the creation of deeper networks without encountering the problem of vanishing gradients. The model uses bottleneck blocks (Fig. 6), which reduce the number of parameters while maintaining processing capability. Each bottleneck block comprised three convolution layers: a 1 × 1 convolution to reduce dimensionality, a 3 × 3 convolution for feature extraction, and a 1 × 1 convolution to restore dimensionality. The bottleneck blocks make ResNet50 more parameter-efficient, enabling it to process high-resolution images more effectively.

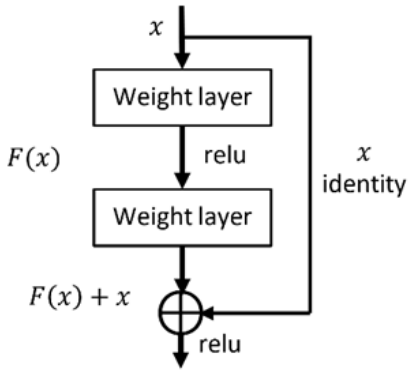

Figure. 5 Residual learning: a building block Redrawn from [31]

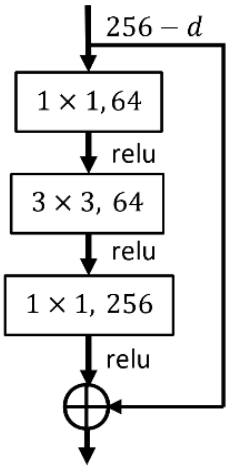

Figure. 6 A "bottleneck" building block for ResNet-50/101/152 Redrawn from [31]

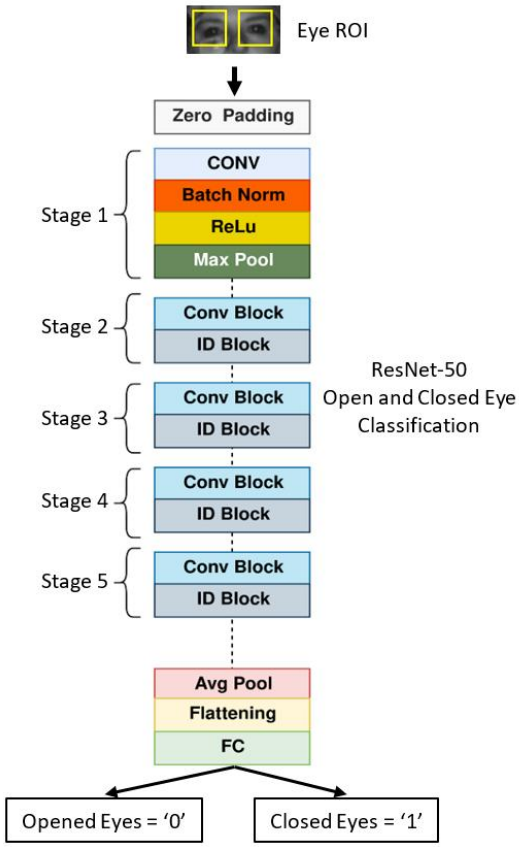

Figure. 7 Blink Detection Block Diagram Redrawn from [32]

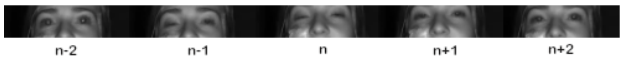

Figure. 8 Blinking Pattern

A key feature of ResNet50 (Fig. 5) is the use of skip connections, or shortcut connections, in each residual block. In each block, the original input is added to the convolution output before proceeding to the next layer (Eq. (7)). Mathematically, this can be expressed as:

$$H(x) = F(x, W) + x \tag{7}$$

where *F(x,W)* represents the residual mapping to be learned as a nonlinear function representing the convolution operation with layer weights *W*, and *x* is the shortcut connection that adds the original input *x* to the result of *F(x,W)*. This approach enables the model to learn the residual, or the difference between the input and output, which aids in training the network and maintaining stability while ensuring that important information is retained as it flows through the network's blocks.

To prevent vanishing gradients, ResNet-50 allows gradients to flow directly through the shortcut paths without passing through all convolution layers. When the image resolution or number of features changes between layers, 1 X 1 convolutions are used in the shortcut connections to ensure that the input and output have matching dimensions.

The ResNet architecture is widely used for object processing and recognition in two-dimensional data [32, 33]. In this study, we implement CNN ResNet-50 to classify whether the eyes are open or closed in each video frame as shown in Fig. 7. The open and closed right and left eye datasets were trained with two classes: open and closed. Thus, the image data at frames *(n – 2)* and *(n + 2)* in Fig. 8 represent open eyes, while the image data at frames *(n – 1)*, *(n)*, and *(n + 1)* represent closed eyes.

The classification of open and closed eyes was based on the open and closed eye dataset generated from the blink data series of the Drozy dataset. The sequence of eye images in the blink flow was divided into two classes: open and closed eyes. The dataset was trained using ResNet-50, and the same architecture model was employed for classifying open and closed eyes. Images of detected open eyes were classified as '0', while closed eye images were classified as '1'.

Open/Closed Eye Filter: Eye closure should not automatically be interpreted as a sign of tiredness or drowsiness [34]. There are various scenarios where eye closure occurs for different reasons. For instance, an eye roll may be used as part of social interaction and should not be misinterpreted as a sign of drowsiness. Similarly, eyes that appear to be grinning should not be mistaken for a wink. Additionally, spontaneous blinking can result from multiple factors [35], such as dry eye conditions [36], or a decline in physical well-being, which could be related to fatigue or drowsiness. It is essential to recognize that a typical blink involves the brief or extended closure of

both eyelids, whereas squinting is a deliberate action, often used as a specific facial expression or a form of communication. Therefore, the nuances of eye movements and conditions should be carefully considered, especially when using them as potential indicators of one's physical or emotional state.

In this study, blink detection is based on the condition of the eyelids of both eyes (right eye $E_r(n)$ and left $E_l(n)$), which must be closed in the same frame image $I(n)$, as shown in Figure . Separate detection of the right eye $E_r(n)$ and left eye $E_l(n)$ is designed to ensure valid information regarding eye closure. The classification of closed or open eyes in a frame image $I(n)$ follows the rules illustrated in Fig. 9 and Eq. (8). An eye detected as closed is assigned a logic value of '1', while an open eye is assigned a logic value of '0'. The eye features $f(n)$ in the frame image $I(n)$ will be classified as closed if and only if both the right eye $E_r(n)$ and the left eye $E_l(n)$ are detected as closed.

$$f(n) = \begin{cases} 1, if\ E_l(n) = 1 \wedge E_r(n) = 1 \\ 0, others \end{cases} \quad (8)$$

Open/Closed Pattern: The pattern of eye opening and closing, or blinking, serves as a key feature in assessing an individual's level of alertness. However, it should be noted that blinking can also indicate certain eye health issues. For instance, the frequency of blinking increases when the eyes are dry [36], or when an individual is experiencing fatigue or drowsiness. As fatigue or sleepiness increases, blink frequency and duration tend to rise. Therefore, blinking patterns are valuable indicators for gauging focus, fatigue, or drowsiness [37] that utilize camera sensors have been developed to capture the subtle features of eye blinks within a sequence of facial images. As outlined in studies [38] and [35], a single blink period is characterized by a specific sequence of eye states: transitioning from open $(n-2)$, to semi-closed $(n-1)$, fully closed (n), semi-open $(n+1)$, and back to open $(n+2)$, as shown in Fig. 8.

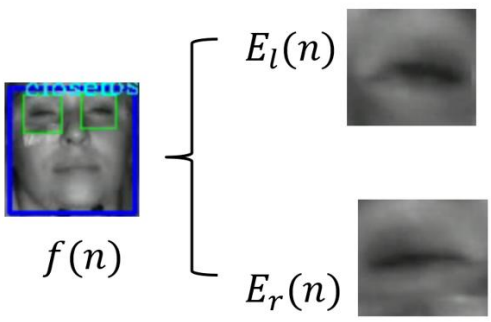

Figure. 9 Detection of right and left eye features

Figure. 10 One-period Binary Blink Pattern

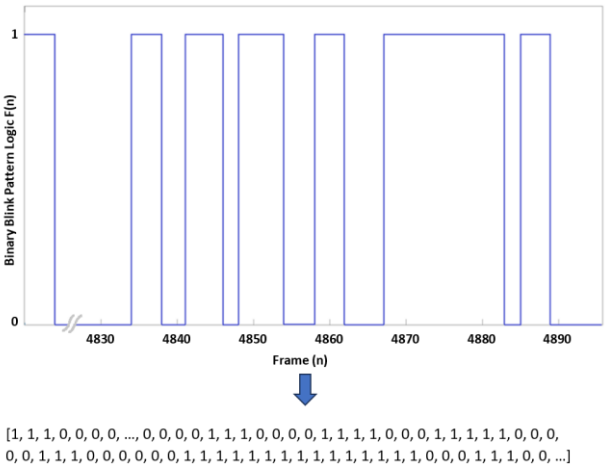

Figure. 11 Binary Blink Pattern Illustration

In Fig. 8, one blink period is defined as a change in state from open eyes to closed eyes and then back to the open state, while the transition conditions between open and closed eyes are ignored. Referring to Eq. (8) regarding the binary classification of open and closed eyes, a blink period is detected when the eye state changes from '0' (open) to '1' (closed) and back from '1' to '0'.

As illustrated in Fig. 10, the blink pattern for one period is "01110". Fig. 11 shows a binary blink pattern over a single time duration. From this pattern, it is possible to calculate the number of blinks and the duration of each blink in each period.

The duration of eye closure within a single blink correlates with the length of the '1' logic in the binary data pattern, which indicates increased drowsiness. Over a span of 1 to 10 minutes, variations in blink patterns and eye closure durations serve as indicators of drowsiness or alertness. These patterns provide reference data for statistical analysis of blinks, which are used to train models that classify drowsiness or alertness. A frequent and prolonged binary blink pattern suggests a predominance of eye closure, indicating drowsiness, while a less frequent and shorter pattern suggests alertness.

3.4 Blinking feature extraction

The feature extraction stage is designed to obtain unique blink frequency and duration patterns within a single time period from the binary data series. This binary blink data series represents serial data for a single video duration, which is interpreted into a binary blink pattern matrix to derive the statistical

features of blink frequency and duration. The interpretation results in unique pattern features of blink frequency and duration, which are then used as training and test data. Several computational stages are performed to extract these features:

Windowing: To ensure data continuity and integrity, and to accurately analyze features related to eye blinking, the blinking sequences are divided into several segments with a consistent duration. By segmenting the blinking dynamics, metrics such as blink frequency, blink duration, and eye closure duration can be analyzed over specific ranges and steps.

The proposed segmentation method provides a comprehensive view of the drowsiness scale status, allowing for the creation of a unique pattern matrix of blinking behavior. To enhance analysis accuracy, windowing was applied to limit the observation area of the input data, with unobserved areas initially set to '0'. In this study, 1800 data frames were observed for each window length *L*, corresponding to an observation period of 1 minute at 30 fps. This duration was chosen with the assumption that the blinking pattern features could be captured adequately within this time frame. A window step (ϑ) was applied to address data discontinuity due to windowing (Fig. 12). Several variations in the overlap step width were tested to find the optimal overlap value, ensuring the expected classification accuracy. A wider window step affects computation time as more iterations are performed. The binary eye-blink pattern in frame *f(n)* is shown in Fig. 13.

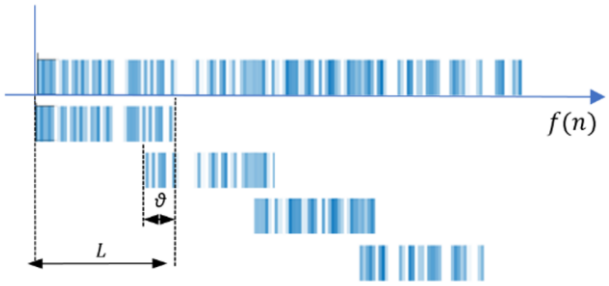

Figure. 12 Window Length and Window Step Data Training

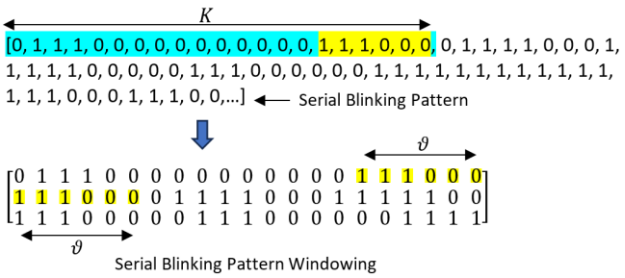

Figure. 13 Blinking Pattern Window

Blinking Frequency and Duration: Drowsiness classification often relies on the analysis of eye blink behavior, as changes in blink patterns are strong indicators of an individual's level of drowsiness. As individuals begin to feel drowsy, their blink patterns change—blinks become slower and longer, resulting in increased blink duration compared to when they are fully alert. Additionally, individuals experiencing drowsiness tend to blink more frequently, with greater variability in blink duration [39].

Blink duration and frequency features are obtained by identifying the indices of elements in a window whose value is '0' in Eq. 3. Specifically, the index of the first frame with a value of '0' is denoted as *f(n)*, and the index of the last frame with a value of '0' is denoted as $f(n_{last})$. The duration of each blink is calculated as the difference between these indices, represented as $D(m) = n_{last} - n$. If no consecutive '1' values are found, the duration is assigned a value of '1' and stored in the duration parameter *D(m)* according to Eq. (9). The frequency of eye blinks in a data window is determined by the number of *D* values greater than '1'. Therefore, if all values in *D* within a window are '1', this indicates that no eye blinks occurred during that window period.

$$D(m) = \sum_{n=0}^{L-1} \begin{cases} while\ f(n) = 0, n_{last} = n + 1, \\ \therefore D(m) = n_{last} - n \\ D(m) = 1 \end{cases} \quad (9)$$

Statistical Feature: The use of statistical features in image-based analysis and classification tasks is widely applied in several fields, including medical case analysis such as liver cancer [40], Parkinson's disease [41], and EEG signal-based drowsiness detection [42]. In this study, drowsiness features are extracted through the statistical analysis of blink duration and frequency within a specific window period, serving as the drowsiness classification feature. The statistical features calculated include:

Mean Distance (μ) (Eq. 10): The average distance between frames detected during the blink period, normalized to the window length *L*. d_i is the blink duration data, a value of '1' indicates no blinks occurred during the data period. *M* is the number of observed data. This metric provides an overview of how frequently eye blinks occur in a window. Normalization by window size and factor 4 allows for comparison across windows of different sizes and adjusts the measurement to a consistent scale.

$$\mu = \frac{\sum_{i=1}^{M} d_i}{M} \times \frac{c}{L} \quad (10)$$

The distance variance *(σ^2)* between blinks measures how much the blink intervals deviate from the mean (Eq. 11). Understanding the spread of blink intervals is crucial for assessing the consistency of the eye blink pattern. A high variance may indicate instability or significant changes in the blink pattern.

$$\sigma^2 = \frac{\sum_{i=1}^{M}(d_i-\mu)^2}{M} \times \frac{c}{L} \quad (11)$$

The standard deviation *(σ)* of blink distance measures how much the intervals between blinks deviate from the mean distance (Eq. 12). This metric is essential for understanding the variability in eye blink patterns, which can provide insights into levels of drowsiness. Normalization based on window size and a positive integer *c* factor (ex: 2, 4, 8, ...) allows for comparison between windows of different sizes and ensures that measurements are adjusted to a specific scale.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{M}(d_i-\mu)^2}{M}} \times \frac{\sqrt{c}}{\sqrt{L}} \quad (12)$$

The algorithm used to reconstruct the dataset based on the statistical features of binary eye blink patterns is shown in Algorithm 1. This algorithm outlines the steps required to identify eye blink patterns, calculate statistical features such as mean, variance, and standard deviation of blink intervals, and reconstruct the dataset using these features. With this approach, the resulting dataset will be more representative of eye blink patterns, making it suitable for further analysis. The *`ExtractFeatures`* function extracts statistical features from time series data using the moving window method. The function accepts three parameters: *`data`* (time series data), *`step`* (window shift step), and *`window_size`* (data window size).

The function begins by initializing an empty list *`windows`*. It then loops through the *`data`* from index *0* to the final index, with each iteration producing a window of size *`window_size`* from the *`data`*, shifting by *`step`* with each iteration. This process populates the `windows` list with the feature extraction results. In each iteration, a data window is extracted from indices *`i`* to *`1 + window_size`*.

Algorithm 1. Statistical Feature Extraction

1. FUNCTION ExtractFeatures(data, step, window_size):
2. INITIALIZE an empty list called 'windows'
3. FOR each index 'i' from 0 to (length of data - window_size + 1) with step size:
   - **a.** EXTRACT a window of data from index i to (i + window_size)
   - **b.** FIND indices of elements in 'window' that are equal to 0
   - **c.** COMPUTE differences between consecutive indices of zeros
   - **d.** IF no differences found:
   - **e.** SET differences to an array with a single element 0
   - **f.** COMPUTE mean distance between zeros normalized by window size
   - **g.** COMPUTE variance of distances between zeros normalized by window size
   - **h.** COMPUTE standard deviation of distances between zeros normalized by window size
   - **i.** CREATE a tuple (mean_distance, variance_distance, std_dev_distance)
   - **j.** APPEND the tuple to the windows list
4. CONVERT windows list to an array
5. RETURN windows array

Within each data window, indices of elements with a value of *'0'* are identified. The difference between successive indices of *'0'* values is then calculated. If a window contains no *'0'* values, a value of *'1'* is appended to the matrix, as shown in Fig. 14. A *'1'* indicates that no blinking state occurred in the data series, while other values represent blink events with durations corresponding to the data values.

```
[ 1  1  1  1  1  …  1  1 11  1  1  1  1  1  1  1 ]
[ 1 35  1  1  1  …  1  1  1  1 19  1  1 143  1  6 ]
[ 1 19  1  1  1  …  1  10  1 38  1  1 84  1 12 ]
[ 26  1  9  9  1 …  1  7  1  1  1  1  7  1  1  3 ]
[ 1  1 143  1  6 … 24  1  5  1  1  1  1 19  1  1 ]
[ 1  1  1 31  1 … 393  1 106  1 25  1 50  1  1]
```

Figure. 14 Illustration of the calculation of the `distance_between_zeros` parameter from the window data series array

```
[0.03226739 5.29226195 0.10844622]
[0.03617521 6.19322465 0.11731463]
[0.05192982 8.89975685 0.14063157]
[0.05062678 8.57884031 0.13807277]
[0.05128889 8.86878578 0.14038665]
[0.04434028 5.48232151 0.11037634]
```

Figure. 15 Statistical features of blinking pattern

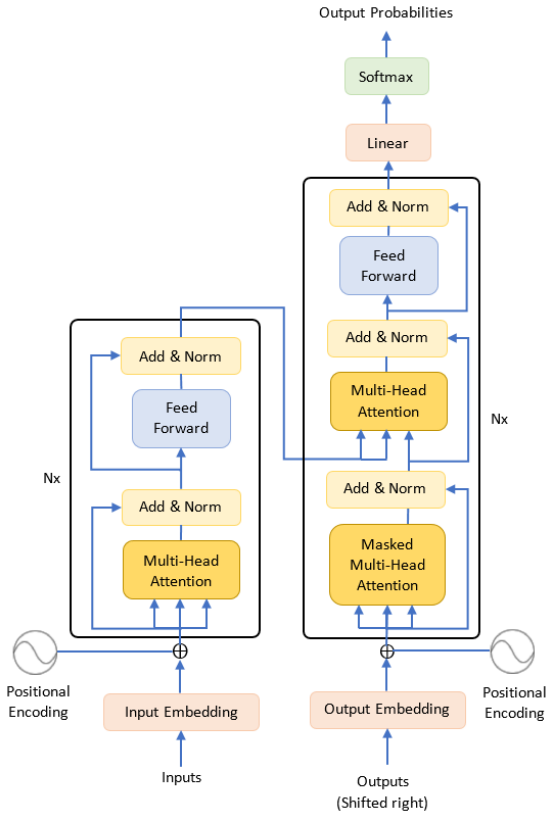

Figure. 16 Time Series Transformer Model. Redrawn from [43]

The statistical features—mean distance, variance, and standard deviation of the distances between *'0'* elements, normalized by the window size—are calculated from these differences. The statistical feature tuples (mean, variance, and standard deviation) are then appended to the *`windows`* list, as shown in Fig. 15. The 1st element in each tuple is the mean distance, the 2nd is the variance, and the 3rd is the standard deviation of the blink pattern features within a single data series window.

The processed windows are compiled into an array and returned as the function's output. In time series data analysis, this function supports the classification of driver drowsiness by identifying patterns in activity changes or physiological parameters. Machine learning models can be trained to detect drowsiness using the statistical features derived from eye blink or heart rate data. The mean, variance, and standard deviation of the distance between eye blinks are critical for identifying drowsiness patterns.

3.5 Drowsiness classification model architecture

The architectural model used for drowsiness classification in this study is based on the Transformer model (Fig. 16) introduced by Vaswani et al. [43]. Originally designed for natural language processing (NLP), the model has since been applied to various tasks, including medical image segmentation [44], hyperspectral image classification [45], and multivariate time series analysis [46]. Unlike earlier recurrent and convolutional architectures, the Transformer relies on an attention mechanism to capture the relationships between elements in the input sequence. The model as shown in Fig. 17 is an encoder-decoder structure, where the encoder converts the input into a continuous representation, and the decoder generates an output based on this representation. The primary mechanism of the Transformer is self-attention, which allows the model to consider the entire input sequence simultaneously. Both the encoder and decoder consist of a stack of N identical layers. Each layer has two sublayers: a multi-head self-attention mechanism and a position-based fully connected simple feed-forward network. Residual connections in each sublayer are followed by layer normalization to address the vanishing gradient problem. Specifically, the output of each sublayer is computed as:

$$LayerNorm(x + Sublayer(x))$$

where $Sublayer(x)$ is the function implemented by the sublayer.

Multi-Head Attention: The Transformer model utilizes multi-head attention, which performs several attention operations in parallel. Each attention head has its own query, key, and value projections. The results from all the heads are combined and projected back to produce the final output [43].

Positional Encoding and Feedforward Networks: Unlike RNN models, the Transformer does not inherently capture the sequential order of the input. To provide the model with information about the relative position of each token in the sequence, positional encoding is used, adding sinusoidal and cosinusoidal signals to the input [43].

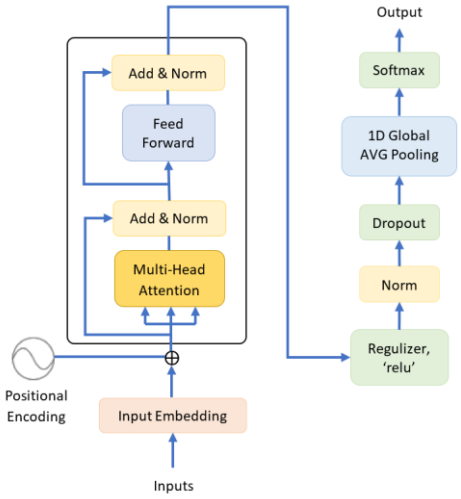

Figure. 17 Proposed Transformer Model

Table 2. Proposed Model Hyperparameter

Hyperparameter	Configuration
Model dim	128
Num. Head	4
Num. layer	3
Kernel Regularizer	*L2*(0.02)
Drop Out Rate	0.4

In some applications of Transformer models, it is not always necessary to use the decoder function [46], [47]. In the standard Transformer architecture, the decoder is typically required for sequence-to-sequence tasks, such as translation, where the model is expected to produce a sequence of outputs. The decoder processes the information generated by the encoder and considers the output sequence that has already been generated (in an auto-regressive setting).

In this study, output sorting was not required because the task involved classification (producing class probabilities for the entire input sequence). Therefore, only the encoder component was necessary to extract features from the input sequence. Model improvements included the addition of a Dense layer with "relu" activation and a kernel regularizer to mitigate the potential for overfitting. The inclusion of 1D Global Average Pooling (GAP) was employed to reduce the temporal dimension of the time series data. After the Multi-Head Attention blocks and dense operations, the data still retained a temporal dimension (number of time steps) and features (dimensionality increased by `*model_dim*`). The kernel regularizer was specifically added to reduce overfitting in the TST model, as shown in Fig. 17. Table 2 presents the hyperparameters proposed for the Transformer model to classify drowsiness levels.

4. Results and discussion

This chapter discusses the research results based on the methodology and model architecture. The testing results were compared with several models that were independently tested and also compared with research conducted by other researchers using the same dataset. Few researchers have used the Drozy video dataset due to its low image resolution. In independent testing, data preprocessing was performed using the statistical features of the blink pattern in the two models. The comparison models were Resnet50 + Long Short Term Memory Network (LSTM) and Resnet50 + CNN 1D. Additionally, a comparison was made with a study by Han et al. [23], which used both spatial and temporal facial features for fatigue identification through multiple modalities. Their model utilized a combination of deep learning and eyelid information to determine operator fatigue. Han et al. combined Temporal Convolutional Networks (TCN) and ResNet3D to preserve both the temporal and spatial features of the eyelid. The model training process was optimized using a custom cosine annealing learning rate decay algorithm to prevent local optimums.

Testing was performed using training data with a windowing duration of 1,800 frames of input data, applied to each model. The model with the best results is illustrated in Fig. 18. From the model accuracy comparison graph, it can be observed that all three models (ResNet50 + TST, ResNet50 + LSTM, and ResNet50 + CNN 1D) show an increase in accuracy for both training and validation data as the number of epochs increases. The ResNet50 + TST and ResNet50 + CNN 1D models demonstrated more consistent performance, with stable validation accuracy approaching 0.97–0.98 by the end of training. In contrast, the ResNet50 + LSTM model exhibited lower and more fluctuating validation accuracy, particularly during the early and middle epochs, indicating that it was more prone to overfitting than the other two models.

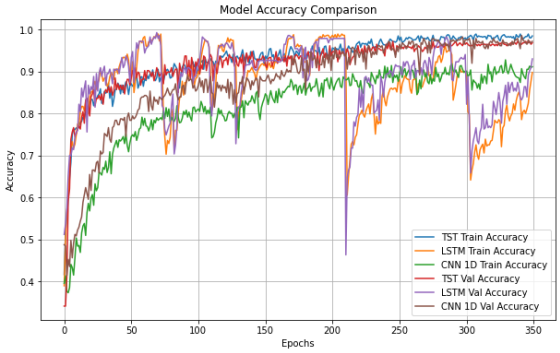

Figure. 18 Comparison of training and validation accuracy for ResNet50 + LSTM, ResNet50 + CNN-1D, and ResNet50 + TST models.

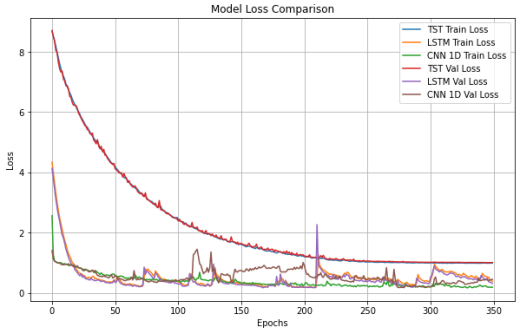

Figure. 19 Comparison of training and validation loss for ResNet50 + LSTM, ResNet50 + CNN-1D, and ResNet50 + TST models.

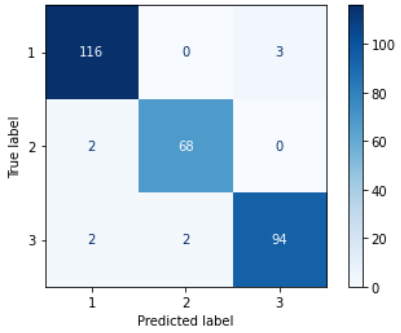

Figure. 20 Confusion Matrix of Resnet50 + TST

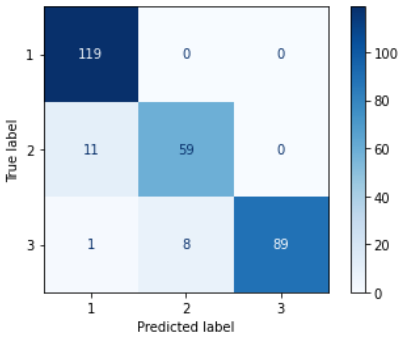

Figure. 21 Confusion Matrix of Resnet50 + LSTM

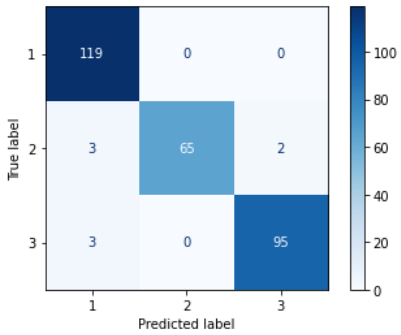

Figure. 22 Confusion Matrix of Resnet50 + CNN 1D

The loss graph in Fig. 19. reinforces this observation, showing that the ResNet50 + LSTM model experienced a slower reduction in loss compared to the ResNet50 + TST and ResNet50 + CNN 1D models. Both ResNet50 + TST and ResNet50 + CNN 1D displayed steeper loss curves, indicating that these models learned patterns from the data more effectively. This is evidenced by the loss values approaching zero by the end of training, especially for the training data. However, on the validation data, the ResNet50 + CNN 1D model demonstrated slightly more stability than the ResNet50 + TST, although both models still outperformed the ResNet50 + LSTM. The ResNet50 + LSTM model showed significant fluctuations around epoch 200, suggesting difficulties in maintaining good generalization on the validation data.

Technically, these results demonstrate that the ResNet50 + TST and ResNet50 + CNN 1D architectures are better suited to handle the temporal complexity of the data while avoiding overfitting. Although ResNet50 + LSTM is known to perform well on sequence data, it struggled to identify the correct patterns in this dataset, affecting both the accuracy and loss performance on the validation data. Of the two best-performing models, ResNet50 + TST may have a slight advantage in capturing more complex temporal patterns, while ResNet50 + CNN 1D offers better stability with faster training times.

Based on the confusion matrix of the three models (Figs. 20-22), the ResNet50 + TST model demonstrated the best performance, with the fewest classification errors. ResNet50 + TST, utilizing the self-attention mechanism, effectively captured the long-term relationships in the statistical features of the eye blink pattern (mean, variance, and standard deviation). There were only a few misclassifications: 2 instances of class 2 were classified as class 1, and 2 instances of class 3 were classified as class 2. This highlights ResNet50 + TST's ability to accurately distinguish between different levels of drowsiness, with minimal errors, due to its strength in fully integrating temporal information.

In contrast, the ResNet50 + LSTM model showed the poorest performance among the three, with substantial errors, particularly in class 2. Specifically, 11 instances were misclassified as class 1, and 8 instances of class 3 were classified as class 2. This suggests that although ResNet50 + LSTM was designed to handle sequential data, it struggled to differentiate patterns between classes that are harder to distinguish, such as class 2 and class 3.

ResNet50 + 1D CNN showed relatively good results and was more stable than ResNet50 + LSTM, although there were some prediction errors in class 2. ResNet50 + 1D CNN was better at capturing local spatial patterns than ResNet50 + LSTM, but it was not as effective as ResNet50 + TST in capturing long-term temporal dependencies. While ResNet50 + 1D CNN's performance approached that of TST, it did not achieve the same level of precision in distinguishing between classes with similar pattern variations.

Based on the analysis of confusion matrix, accuracy, and loss, the ResNet50 + TST and ResNet50 + CNN 1D models demonstrate excellent performance in drowsiness classification compared to ResNet50 + LSTM. The ResNet50 + TST model leverages the self-attention mechanism, which excels in capturing long-term temporal dependencies in eye blink statistical features (mean, variance, and standard deviation). This results in stable validation

accuracy above 0.97 and consistently low loss, reflecting the model's ability to avoid overfitting and capture complex patterns across classes. ResNet50 + CNN 1D, which applies one-dimensional convolution, is effective in capturing local patterns from sequential data, achieving performance nearly on par with ResNet50 + TST. It exhibits high validation accuracy and stable loss, though it is slightly less capable of capturing long-term dependencies.

Figs 18 and 19 shows the performance of ResNet50 + TST (blue and red) versus ResNet50 + CNN 1D (green and brown). The difference in training accuracy - validation accuracy and training loss - validation loss for ResNet50 + TST is smaller than that of ResNet50 + CNN 1D. This indicates that the ResNet50 + TST model has better generalization capabilities and is less prone to overfitting compared to ResNet50 + CNN 1D.

In contrast, ResNet50 + LSTM exhibited weaknesses in capturing pattern variations between classes, particularly in class 2, as reflected in its higher classification error and large fluctuations in the accuracy and loss graphs. ResNet50 + LSTM experienced overfitting, as indicated by the increase in validation loss after the 200th epoch, which resulted in poorer generalizability to the validation data compared to ResNet50 + TST and ResNet50 + CNN 1D. Table 3 shows that the combination of ResNet50 + TST and ResNet50 + 1D CNN yielded equally strong results in terms of accuracy (0.97), recall (0.97), and F1-score (0.97), indicating that both models are highly effective in detecting drowsiness patterns based on the statistical features of eye blink patterns. However, ResNet50 + 1D CNN slightly outperforms ResNet50 + TST in terms of precision (0.98 compared to 0.97), meaning that ResNet50 + 1D CNN is marginally better at minimizing false positives. In contrast, ResNet50 + LSTM demonstrated lower performance, with an accuracy of 0.93 and an F1-score of 0.92. This suggests that ResNet50 + LSTM is less effective in handling complex patterns and long-term dependencies compared to ResNet50 + TST and ResNet50 + 1D CNN, which excel at capturing temporal and spatial patterns, respectively.

Figs. 18 and 19 (The accuracy and The loss graphs) and Figs. 20-22 (confusion matrix) provide a comprehensive overview of the model performances in drowsiness classification based on the statistical features of eye blink patterns. The accuracy and loss graphs illustrate how each model performed during training and validation. ResNet50 + TST and ResNet50 + 1D CNN demonstrated excellent and stable accuracy trends, particularly after 150 epochs,

Table 3. Comparison of Classification Reports for Different Models

Model	Accuracy	Precision	Recall	F1-Score
Resnet 50 + LSTM	0.93	0.93	0.92	0.92
Resnet 50 + 1DCNN	0.97	0.98	0.97	0.97
Resnet 50 + TST	0.97	0.97	0.97	0.97

maintaining near-perfect validation accuracy (around 0.97). In contrast, ResNet50 + LSTM exhibited large fluctuations, especially after the 200th epoch, indicating that ResNet50 + LSTM struggled to achieve stability during training, especially when generalizing to validation data.

From the loss graph in Fig. 19, it can be seen that ResNet50 + TST and ResNet50 + 1D CNN quickly achieved a low and stable loss, while ResNet50 + LSTM experienced fluctuations, particularly on the validation data. This reflects ResNet50 + LSTM's difficulty in adapting to complex patterns, making it less suitable for this classification task compared to the other models.

The Accuracy graph (Fig. 18), the loss graph (Fig. 19), and the confusion matrix (Figs. 20-22) also conclude that statistical feature-based data preprocessing provides a solid foundation for the model to learn drowsiness patterns; however, the effectiveness of each model in optimizing these features differs. ResNet50 + TST, with its self-attention mechanism, excels at capturing complex temporal patterns, while ResNet50 + CNN 1D is also highly effective due to its convolutional architecture, which can recognize local spatial patterns. ResNet50 + LSTM, although expected to perform well with sequential data, struggles to capture long-term patterns, as evidenced by its lower accuracy and performance in the confusion matrix. Therefore, ResNet50 + TST and ResNet50 + CNN 1D are better equipped to optimize the statistical features of eye blinks for drowsiness classification than ResNet50 + LSTM.

Table 4 compares the accuracy of different drowsiness detection and classification methods based on the type of features used. Facial landmark-based approaches, such as VBFLLFA Transformer (93.10%) and 2s-STGCN (92.2%), demonstrate good ability to exploit spatial information from faces, but tend to be limited in capturing complex temporal patterns. The eyelid aspect ratio (EAR)-based methods, such as SVM, achieved 94.9% accuracy,

Table 4. Accuracy comparison for different models and dataset features

Features	Method	Accuracy (%)
Facial Landmark and local Area [17]	VBFLLFA Transformer	93.10
Facial Landmark [22]	2s-STGCN	92.2
EAR Sequence [20]	SVM	94.9
EAR Sequence [23]	1D CNN	46.25
	LSTM128	42.92
	LSTM256	43.33
	TCN	51.11
Both Eyelid Image and EAR Sequence [23]	Resnet3D +1DCNN	97.36
	Resnet3D +TCN	97.64
Eyelid Image + Statistical binary blink (our Method)	Resnet50 +1DCNN	97
	Resnet50 +LSTM	63
	Resnet50 +TST	97

demonstrating the effectiveness of simple approaches for sequential data. However, deep learning methods such as 1D CNN (46.25%), LSTM (42.92%-43.33%), and TCN (51.11%), perform much less effectively, which reflects their limitations relative to exploiting EAR data.

Multimodal approaches such as ResNet3D + 1DCNN (97.36%) and ResNet3D + TCN (97.64%) achieve high accuracy due to their ability to capture both spatial and temporal information from the combination of eyelid and EAR images. However, this method has the disadvantage of complexity and dependence on two types of data (multimodality). In contrast, the proposed ResNet50 + TST method uses only one type of data (single modality) in the form of eyelid images and statistical binary blink features, but still achieves the same high accuracy of 97%. By using a single modality, the proposed method is not only architecturally simpler and computationally more efficient but also more robust to environmental variability than multimodal approaches such as ResNet3D + TCN. The results demonstrate the adnvantage of the proposed method in handling drowsiness detection without requiring additional complexity from multimodal integration.

5. Conclusion

The proposed ResNet50 + TST method demonstrates effectiveness in detecting and classifying drowsiness using statistical binary blink features derived from eye image data. Achieving an accuracy of 97%, the proposed method performs comparably to multimodal approaches, such as ResNet3D + TCN, but with reduced complexity because it relies on a single modality. The integration of Time Series Transformer (TST) enables efficient modeling of temporal dependencies, and ResNet50 ensures reliable spatial feature extraction. These advantages make the proposed method more robust to environmental variations, such as inconsistent lighting and head movements, and reduce the reliance on high-precision facial landmarks.

Compared to other multimodal methods, the ResNet50 + TST framework offers a simpler and more computationally efficient solution without sacrificing accuracy. This makes it particularly suitable for real-time applications in which computational resources are limited. Furthermore, the proposed method maintains high accuracy even when trained and tested on low-resolution data (8-bit, 512 × 424), demonstrating its flexibility and adaptability. These characteristics highlight the potential of the proposed method for application to diverse scenarios and datasets that require reliable and efficient drowsiness detection. Overall, the ResNet50 + TST approach is a practical and effective solution for vision-based systems to detect and manage drowsiness in real-world environments.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, writing—original draft preparation, writing—review and editing, visualization, have been done by 1st author. The supervision and writing—review, have been done by 2nd and 4th Author, and project administration have been done by 3rd author.

Acknowledgments

We would like to thank the University of Liège's research team for providing access to the Drowsiness dataset.

References

[1] *Global Status Report on Road Safety 2023*. Geneva: World Health Organization, pp. 64, 2023.

[2] J. Skorucak, A. Hertig-Godeschalk, D. R. Schreier, A. Malafeev, J. Mathis, and P. Achermann, "Automatic detection of microsleep episodes with feature-based machine

learning", *Sleep, International Journal For Sleep And Circadian Science*, Vol. 43, No. 1, 2020, doi: 10.1093/sleep/zsz225.

[3] G. R. Poudel, C. R. H. Innes, P. J. Bones, R. Watts, and R. D. Jones, "Losing the struggle to stay awake: Divergent thalamic and cortical activity during microsleeps: Neural Activity During Microsleeps", *International Journal of Human Brain Mapping*, Vol. 35, No. 1, pp. 257-269, 2014, doi: 10.1002/hbm.22178.

[4] A. Balaji, U. Tripathi, V. Chamola, A. Benslimane, and M. Guizani, "Toward Safer Vehicular Transit: Implementing Deep Learning on Single Channel EEG Systems for Microsleep Detection", *IEEE Trans. Intell. Transport. Syst.*, Vol. 24, No. 1, pp. 1052-1061, 2023, doi: 10.1109/tits.2021.3125126.

[5] C. Wang *et al.*, "Spectral Analysis of EEG During Microsleep Events Annotated via Driver Monitoring System to Characterize Drowsiness", *IEEE Trans. Aerosp. Electron. Syst.*, Vol. 56, No. 2, pp. 1346-1356, 2020, doi: 10.1109/TAES.2019.2933960.

[6] A. Chowdhury, R. Shankaran, M. Kavakli, and Md. M. Haque, "Sensor Applications and Physiological Features in Drivers' Drowsiness Detection: A Review", *IEEE Sensors J.*, Vol. 18, No. 8, pp. 3055-3067, 2018, doi: 10.1109/jsen.2018.2807245.

[7] S. Kaplan, M. A. Guvensan, A. G. Yavuz, and Y. Karalurt, "Driver Behavior Analysis for Safe Driving: A Survey", *IEEE Trans. Intell. Transport. Syst.*, Vol. 16, No. 6, pp. 3017-3032, 2015, doi: 10.1109/tits.2015.2462084.

[8] Y. Saito, M. Itoh, and T. Inagaki, "Bringing a Vehicle to a Controlled Stop: Effectiveness of a Dual-Control Scheme for Identifying Driver Drowsiness and Preventing Lane Departures Under Partial Driving Automation Requiring Hands-on-Wheel", *IEEE Trans. Human-Mach. Syst.*, Vol. 52, No. 1, pp. 74-86, 2022, doi: 10.1109/thms.2021.3123171.

[9] J. Zhang and N. Xiao, "EEG and Forehead EOG-Based Driver Fatigue Classification Using Sparse-Deep Belief Networks", In: *Proc. of IEEE 23rd Int Conf on High Performance Computing & Communications*, Haikou, Hainan, China, 2021, doi: https://doi.org/10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00166.

[10] H. Martensson, O. Keelan, and C. Ahlstrom, "Driver Sleepiness Classification Based on Physiological Data and Driving Performance From Real Road Driving", *IEEE Trans. Intell. Transport. Syst.*, Vol. 20, No. 2, pp. 421-430, 2019, doi: 10.1109/TITS.2018.2814207.

[11] A. R. Ismail, D. Sodoyer, and F. Elbahhar, "Drowsiness Detection in Humans based on ECG Analysis Using Temporal Convolutional Network", In: *Proc. of 2023 International Conference on Automation, Control and Electronics Engineering (CACEE)*, Chongqing, China, pp. 62-66, 2023, doi: 10.1109/cacee61121.2023.00021.

[12] I. A. Akbar, T. Igasaki, N. Murayama, and Z. Hu, "Drowsiness assessment using electroencephalogram in driving simulator environment", In: *Proc. of 2015 8th International Conference on Biomedical Engineering and Informatics (BMEI)*, Shenyang, China, pp. 184-188, 2015, doi: 10.1109/BMEI.2015.7401497.

[13] N. F. Hikmah, R. Setiawan, and M. D. Gunawan, "Sleep Quality Assessment from Robust Heart and Muscle Fatigue Estimation Using Supervised Machine Learning", *IJIES*, Vol. 16, No. 2, pp. 319-331, 2023, doi: 10.22266/ijies2023.0430.26.

[14] L. Boon-Leng, L. Dae-Seok, and L. Boon-Giin, "Mobile-based wearable-type of driver fatigue detection by GSR and EMG", In: *Proc. of 2015 IEEE Region 10 Conference*, Macao, 2015, doi: 10.1109/tencon.2015.7372932.

[15] M. H. Baccour, F. Driewer, E. Kasneci, and W. Rosenstiel, "Camera-Based Eye Blink Detection Algorithm for Assessing Driver Drowsiness", In: *Proc. of 2019 IEEE Intelligent Vehicles Symposium (IV)*, Paris, France, pp. 987-993, 2019, doi: 10.1109/ivs.2019.8813871.

[16] S. Al-gawwam and M. Benaissa, "Robust Eye Blink Detection Based on Eye Landmarks and Savitzky-Golay Filtering", *Information*, Vol. 9, No. 4, p. 93, 2018, doi: 10.3390/info9040093.

[17] L. Yang, H. Yang, H. Wei, Z. Hu, and C. Lv, "Video-Based Driver Drowsiness Detection With Optimised Utilization of Key Facial Features", *IEEE Trans. Intell. Transport. Syst.*, Vol. 25, No. 7, pp. 6938-6950, 2024, doi: 10.1109/TITS.2023.3346054.

[18] Q. Cheng, W. Wang, X. Jiang, S. Hou, and Y. Qin, "Assessment of Driver Mental Fatigue Using Facial Landmarks", *IEEE Access*, Vol. 7, pp. 150423-150434, 2019, doi: 10.1109/ACCESS.2019.2947692.

[19] S. Sathasivam, A. K. Mahamad, S. Saon, A. Sidek, M. M. Som, and H. A. Ameen, "Drowsiness Detection System using Eye Aspect Ratio Technique", In: *Proc. of 2020 IEEE Student Conference on Research and*

Development (SCOReD), Batu Pahat, Malaysia, pp. 448-452, 2020, doi: 10.1109/SCOReD50371.2020.9251035.

[20] C. B. S. Maior, M. J. D. C. Moura, J. M. M. Santana, and I. D. Lins, "Real-time classification for autonomous drowsiness detection using eye aspect ratio", *Expert Systems with Applications*, Vol. 158, p. 113505, 2020, doi: 10.1016/j.eswa.2020.113505.

[21] Jian-Feng Xie, Mei Xie, and Wei Zhu, "Driver fatigue detection based on head gesture and PERCLOS", In: *Proc. of 2012 International Conference on Wavelet Active Media Technology and Information Processing (ICWAMTIP)*, Chengdu, China, pp. 128-131, 2012, doi: 10.1109/ICWAMTIP.2012.6413456.

[22] J. Bai *et al.*, "Two-Stream Spatial-Temporal Graph Convolutional Networks for Driver Drowsiness Detection", *IEEE Trans. Cybern.*, Vol. 52, No. 12, pp. 13821-13833, 2022, doi: 10.1109/TCYB.2021.3110813.

[23] X. Han and L. Dai, "Multimodal Fatigue Recognition State Based on Eyelid Features", In: *Proc. of 2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT)*, Jilin, China, pp. 856-865, 2023, doi: 10.1109/ICCECT57938.2023.10141258.

[24] Y. Kudo and A. Kubota, "Image dehazing method by fusing weighted near-infrared image", In: *Proc. of 2018 International Workshop on Advanced Image Technology (IWAIT)*, Chiang Mai, pp. 1-2, 2018, doi: 10.1109/IWAIT.2018.8369744.

[25] Q. Massoz, T. Langohr, C. Francois, and J. G. Verly, "The ULg multimodality drowsiness database (called DROZY) and examples of use", In: *Proc. of 2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Placid, NY, USA, pp. 1-7, 2016, doi: 10.1109/WACV.2016.7477715.

[26] R. Ghoddoosian, M. Galib, and V. Athitsos, "A Realistic Dataset and Baseline Temporal Model for Early Drowsiness Detection", In: *Proc. of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Long Beach, CA, USA, pp. 178-187, 2019, doi: 10.1109/CVPRW.2019.00027.

[27] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", In: *Proc. of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Kauai, HI, USA, p. I-511-I-518, 2001, doi: 10.1109/CVPR.2001.990517.

[28] R. Khilkhal and M. Ismael, "Brain Tumor Detection in MRI Images Using Histogram of Oriented Gradient Features", In: *Proc. of 2022 2nd International Conference on Advances in Engineering Science and Technology (AEST)*, Babil, Iraq, pp. 704-709, 2022, doi: 10.1109/AEST55805.2022.10412889.

[29] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", In: *Proc. of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, pp. 886-893, 2005, doi: 10.1109/CVPR.2005.177.

[30] O. Déniz, G. Bueno, J. Salido, and F. De La Torre, "Face recognition using Histograms of Oriented Gradients", *Pattern Recognition Letters*, Vol. 32, No. 12, pp. 1598-1603, 2011, doi: 10.1016/j.patrec.2011.01.004.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", In: *Proc. of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, doi: 10.1109/cvpr.2016.90.

[32] A. Shoukat, S. Akbar, S. A. Hassan, S. Iqbal, A. Mehmood, and Q. M. Ilyas, "Automatic Diagnosis of Glaucoma from Retinal Images Using Deep Learning Approach", *Diagnostics*, Vol. 13, No. 10, p. 1738, 2023, doi: 10.3390/diagnostics13101738.

[33] W. Liu, G. Wu, F. Ren and X. Kang, "DFF-ResNet: An insect pest recognition model based on residual networks", *Big Data Mining and Analytics*, Vol. 3, No. 4, pp. 300-310, 2020, doi: 10.26599/BDMA.2020.9020021

[34] T. Jung, S. Kim, and K. Kim, "DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern", *IEEE Access*, Vol. 8, pp. 83144-83154, 2020, doi: 10.1109/access.2020.2988660.

[35] R. Paprocki and A. Lenskiy, "What Does Eye-Blink Rate Variability Dynamics Tell Us About Cognitive Performance?", *Front. Hum. Neurosci.*, Vol. 11, 2017, doi: 10.3389/fnhum.2017.00620.

[36] Y. Su, Q. Liang, G. Su, N. Wang, C. Baudouin, and A. Labbé, "Spontaneous Eye Blink Patterns in Dry Eye: Clinical Correlations", *Invest. Ophthalmol. Vis. Sci.*, Vol. 59, No. 12, p. 5149, 2018, doi: 10.1167/iovs.18-24690.

[37] J. Oh, S.-Y. Jeong, and J. Jeong, "The timing and temporal patterns of eye blinking are dynamically modulated by attention", *Human Movement Science*, Vol. 31, No. 6, pp. 1353-1365, 2012, doi: 10.1016/j.humov.2012.06.003.

[38] W.-H. Lee, J.-H. Woo, and J. M. Seo, "The Method for Visualization and Analysis of Eye-blinking Patterns using Dynamic Vision", In: *Proc. of 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Berlin, Germany, pp. 409-412, 2019, doi: 10.1109/embc.2019.8857089.

[39] F. Safarov, F. Akhmedov, A. B. Abdusalomov, R. Nasimov, and Y. I. Cho, "Real-Time Deep Learning-Based Drowsiness Detection: Leveraging Computer-Vision and Eye-Blink Analyses for Enhanced Road Safety", *Sensors*, Vol. 23, No. 14, p. 6459, 2023, doi: 10.3390/s23146459.

[40] G. L. Kulkarni, S. S. Sannakki, and V. S. Rajpurohit, "Texture feature analysis for the liver cancer diseases using statistical based feature extraction technique", In*: Proc. of 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, pp. 1028-1033, 2020, doi: 10.1109/I-SMAC49090.2020.9243496.

[41] R. Haloi, J. Hazarika, and D. Chanda, "Selection of Appropriate Statistical Features of EEG Signals for Detection of Parkinson's Disease", In: *Proc. of 2020 International Conference on Computational Performance Evaluation (ComPE)*, Shillong, India, pp. 761-764, 2020, doi: 10.1109/ComPE49325.2020.9200194.

[42] V. P. Balam and S. Chinara, "Statistical Channel Selection Method for Detecting Drowsiness Through Single-Channel EEG-Based BCI System", *IEEE Trans. Instrum. Meas.*, Vol. 70, pp. 1-9, 2021, doi: 10.1109/TIM.2021.3094619.

[43] A. Vaswani *et al.*, "Attention is All you Need", In: *Proc. of NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach California USA, pp. 6000-6010, 2017.

[44] X. Zhang and X. Cheng, "A Transformer Convolutional Network With the Method of Image Segmentation for EEG-Based Emotion Recognition", *IEEE Signal Process. Lett.*, Vol. 31, pp. 401-405, 2024, doi: 10.1109/LSP.2024.3353679.

[45] Y. Wu, J. Feng, G. Bai, Q. Gao, and X. Zhang, "Hyperspectral Image Classification Based on Spectrally-Enhanced and Densely Connected Transformer Model", In: *Proc. of IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, Kuala Lumpur, Malaysia, pp. 2746-2749, 2022, doi: 10.1109/IGARSS46834.2022.9883830.

[46] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A Transformer-based Framework for Multivariate Time Series Representation Learning", In: *Proc. of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Virtual Event Singapore, pp. 2114-2124, 2021, doi: 10.1145/3447548.3467401.

[47] H. Jiang, L. Liu, and C. Lian, "Multi-Modal Fusion Transformer for Multivariate Time Series Classification", In: *Proc. of 2022 14th International Conference on Advanced Computational Intelligence (ICACI)*, Wuhan, China, pp. 284-288, 2022, doi: 10.1109/ICACI55529.2022.9837525.