International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 30, No. 6, June, 2025, pp. 1557-1567
Journal homepage: http://iieta.org/journals/isi

Real-Time Drowsiness Detection and Classification with Deep Learning Model

Ritu Malik[1], Meenu Vijarania[1], Meenakshi Malik[2], Ibrahim Aljubayri[3], Chander Prabha[4*], Mohammad Zubair Khan[3]

[1] Department of Computer Science and Engineering, School of Engineering and Technology, K. R Mangalam University, Gurugram 122103, India
[2] Department of Computer Science and Engineering, School of Engineering and Technology, BML Munjal University, Gurugram 122413, India
[3] Department of Computer Science and Information, Taibah University, Madinah 42361, Saudi Arabia
[4] Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura 140401, India

Corresponding Author Email: prbchander@gmail.com

https://doi.org/10.18280/isi.300614

Received: 6 May 2025
Revised: 10 June 2025
Accepted: 21 June 2025
Available online: 30 June 2025

Keywords:
drowsiness, feature selection, filtering, Flémingo, hidden markov model (HMM), Dual-Tree Complex Wavelet Transform, real-time driver monitoring, gray-level co-occurrence matrix

ABSTRACT

Drowsy driving is a major concern for road safety, leading to accidents and fatalities. This paper presents a novel approach called Optimized Dual-Tree Deep Learning (ODT-DL) for real-time drowsiness detection in drivers. The model uses advanced techniques like image preprocessing, feature extraction, and feature selection. It uses Hidden Markov Models for sequence modelling and classification, enabling accurate drowsiness detection. The experimental evaluation of ODT-DL on two benchmark datasets, YAWDD and NTHU-DDD, shows outstanding performance, with accuracy, precision, recall, and F1-Score consistently exceeding 99%. The model's high discrimination capabilities and low false alarm rates ensure reliable detection. Comparative analysis against other machine learning models, such as AlexNet, ResNet, Support Vector Machine, and ensemble methods, highlights the superiority of ODT-DL. The findings suggest the model's practical implications for enhancing road safety by preventing accidents caused by driver drowsiness, with potential applications in vehicle safety systems. The proposed ODT-DL model holds promise for real-world implementation and opens avenues for future developments in road safety technology.

1. INTRODUCTION

Traffic mishaps remain as an issue of concern to almost all countries with hosts of fatalities, injuries as well as property damage. These are caused by factors such driver's error, bad terrain, faulty mechanical problems and even extreme weather conditions [1]. Even present-day infrastructure improvements, legislation, as well as enhancement of sensibility among road users, road crashes continue to persist a significant problem around the world. These accidents do not only result in other types of losses such as physical and psychological health of families, communities, and the overall health care systems [2]. To reduce road accidents, better road standards, traffic policy harmonization, proper road behavior, and improved car technology are implemented, requiring increased alertness and collaboration among governments, organizations, and individuals [3]. Drivers are crucial in road accidents as they dictate behavior and make decisions during the journey. Common causes include hasty, reckless, risky speed, distracted driving, texting, mobile phone use, drunk driving, and fatigue [4]. Traffic violations pose a threat to drivers, passengers, pedestrians, and other road users. To reduce accidents, it's crucial to observe traffic laws, maintain a reasonable distance, and monitor traffic conditions [5]. Education, awareness campaigns, and strict enforcement of laws significantly influence road users' attitudes towards safe driving and reduce accidents caused by driver-related factors. Therefore, fostering a culture of safe and careful driving is crucial for creating safe roads and minimizing accident impacts [6].

Driver drowsiness detection is a crucial technology developed to reduce the risk of fatigued driving, a prevalent cause of road accidents [7]. This technology uses sensors and algorithms to monitor a driver's alertness at the wheel, recording factors like steering wheel movement, swaying, gaze direction, and racial expressions to detect fatigue or distraction, and produce alarms if necessary [8]. Modern vehicles use assistive technologies like lane-keeping assistance and adaptive cruise control to prevent accidents. Drowsiness detection technology is revolutionizing the industry by reducing accidents caused by tired or dozing drivers [9]. Technology in vehicles can reduce accidents and fatalities, but campaigns should be launched to raise awareness about sleepy or reckless driving risks among road users [10].

CNNs are effective in detecting drowsiness in drivers, especially in image and video processing, and are useful for monitoring and supervising drivers' behavior using vision-based data [11]. In drowsiness detection systems, CNNs are used to analyse video inputs from car cameras to identify motion patterns of a driver's face revealing fatigue or drowsiness [12]. CNN-based systems can predict driver fatigue by extracting features and analysing patterns, allowing real-time evaluations of attentiveness through signals like low eyelids, gaze shift, and fast blinking [13]. High-tech systems can alert drivers to drowsiness through notifications, encouraging them to stay awake or pause the car, using CNNs and other sensors [14]. CNNs are crucial in detecting drowsiness due to their efficient processing of large visual data, which helps prevent fatigue-induced accidents [15].

The integration of CNN-based drowsiness detection systems could enhance road safety by reducing drowsy driving risks, but comprehensive driver education and awareness campaigns are needed [16]. Privacy concerns arise as cars' interiors are monitored by cameras, recording drivers, making it crucial to balance safety with privacy rights. CNN-based systems can be influenced by low light conditions and camera resolution, potentially leading to false positives or negatives [17]. CNN-based systems struggle to maintain performance in diverse environments and changing factors, often overlooking variations in driving behavior or cultural practices related to dozing off, highlighting the need for more comprehensive and accurate systems [18]. The system's performance may be compromised by false positives or false negatives, and additional costs may prevent its inclusion in vehicles, potentially making it only accessible in lower-class vehicles or less developed countries [19]. Automating certain functions could potentially lead to reckless driving due to the belief that technology will prevent accidents.

Continuous research is needed to improve the accuracy and reliability of CNN-based drowsiness detection models [20]; policymakers are obligated to establish rigorous safety and social responsibility standards, ensuring respect for privacy and ethics [21]. Driver education and public awareness campaigns are crucial for preventing risky driving practices. CNN-integrated drowsiness detection systems can quickly determine drivers' drowsiness levels and intervene when needed. This application demonstrates CNNs' versatility in visual data analysis and can improve road safety by reducing accidents due to driver fatigue [22]. The contribution of this research lies in the development and evaluation of the ODT-DL integrated with Hidden Markov Model for drowsiness detection in drivers. Several key contributions are highlighted:

1. The study integrates advanced techniques like Cross Guided Bilateral Filter, SWIFT, GLCM feature extraction, Dual-Tree Complex Wavelet Transform with Walsh Hadamard Transform, and the Flémingo feature selection method into the Hidden Markov Model framework. This novel combination enhances the model's ability to extract informative features and effectively detect drowsiness.

2. ODT-DL achieves exceptional accuracy, precision, recall, and F1-Score values, all above 99%, demonstrating its ability to accurately distinguish between alert and drowsy states.

3. ODT-DL maintains low false positive and false negative rates, ensuring that it effectively identifies drowsy drivers while minimizing unnecessary alerts.

4. The findings suggest that ODT-DL has significant practical implications for enhancing road safety by preventing accidents caused by driver drowsiness. The model's high accuracy and low false alarms make it suitable for integration into vehicle safety systems. The contribution of this research lies in the creation of an innovative and highly effective drowsiness detection model, ODT-DL, which has the potential to significantly improve road safety and reduce accidents caused by drowsy driving.

2. PROPOSED METHOD

Optimized Dual-Tree Deep Learning (ODT-DL) is a highly efficient research method for image preprocessing and feature extraction for road safety drowsiness detection. The framework starts with a cross-guided bilateral filter algorithm to improve image quality and preserve key details. Advanced approaches like the Sleep-Wake Image Feature Transformation (SWIFT) algorithm and the Gray-Level Co-occurrence Matrix (GLCM) are used for feature extraction, capturing necessary information for classifying different types of drowsiness states. The Dual-Tree Complex Wavelet Transform incorporates the Walsh-Hadamard Transform for improved feature representation. The Flemingo integrated approach is used for feature selection, minimizing dimensionality. ODT-DL uses a Hidden Markov Model (HMM) to capture temporal driver drowsiness patterns, examining driving dynamics. A ranking-based ADA boosting integrated regression classifier predicts drowsiness state using an ensemble learning model with many weak learners, enhancing road safety. The ODT-DL method is a comprehensive approach for image pre-processing, feature extraction, feature selection, and classification, making it useful in real-life driving conditions, where monitoring driver fatigue is critical for reducing accidents on the road.

2.1 Dataset

The YAWDD (Yet Another Wearable Drowsiness Dataset) and NTHU-DDD (National Tsing Hua University Drowsy Driver Detection) datasets are also included as the effective assets that have been employed in the field of research and development of drowsy driver detection systems. Real-world data have been gathered in these datasets for training and testing purposes of algorithms and models for detecting driver drowsiness.

The YAWDD dataset, a wearable drowsiness measurement tool, uses wearable sensors like EEG, EOG, and EMG to measure brain electrical activity, eye movement, muscle movement, and video data. It is synchronized with other datasets and provides examples of varying drowsiness levels, from alertness to sleep. The data is collected using EEG, EOG, and EMG, allowing for more accurate and comprehensive drowsiness assessments.

NTHU-DDD (National Tsing Hua University Drowsy Driver Detection): NTHU-DDD is extracted from video clips of drivers in different situations; while driving at day and night, in good and in bad weather, and in various Road Traffic Situations. NTHU-DDD contains video frames recorded in-car from cameras for which each frame has a label of whether the driver exhibits drowsy or not. The dataset contains the labels 0 representing the frames as non-drowsy while the label 1 represents drowsy frames. Both datasets are beneficial for researchers and developers in detecting driver drowsiness, enabling the development and evaluation of algorithms and models for improving road safety by providing signals to awaken drivers or autonomous vehicle drivers to prevent fatigue-related accidents.

2.2 Steps in ODT-DL

The proposed ODT-DL method, in general, looks quite complex for drowsy driver detection though it might be just because of the insufficient familiarity with that method. The following are the stages involved this proposed method as shown in Figure 1.

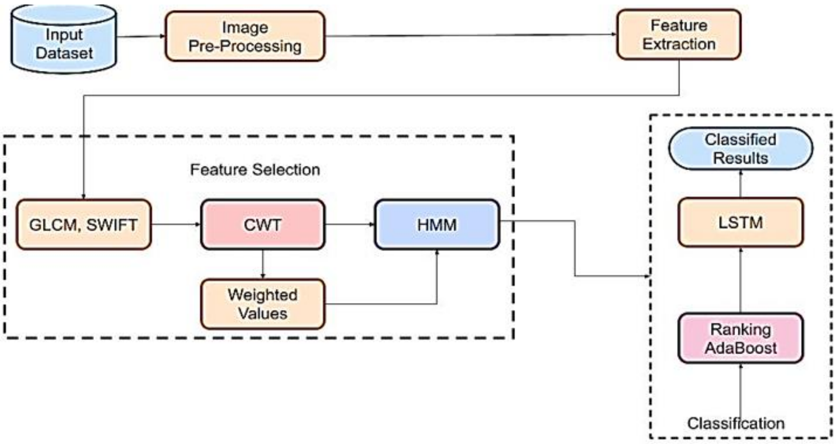


Figure 1. Steps in ODT-DL

This is followed by image preprocessing which will entail making corrections and improvements to the images and data so as to feed the network with optimum images for feature extraction as well as classification. Specifically, one of the preprocessing stages involve the utilization of the Cross Guided Bilateral Filter. This filter is used to work on respective characteristics of the image and minimize the noise in the image [23]. After preprocessing images, specific features are then extracted from the images. These features appear to be essential in helping to separate drowsy from alert conditions. In feature extraction, there is a use of GLCM (Gray-Level Co-occurrence Matrix). GLCM works with pixel and captures spatial relations between them, it can describe texture and pattern presented in the images [24, 25]. Thus, the images are processed using Complex Wavelet Transform (CWT) in order to explore further details of the images. CWT is a useful method in order to obtain the details of image at different resolutions and orientations [26]. Also, there is the Walsh-Hadamard Transform which is carried out. SWIFT (Spherical Wavelet Transform) is a feature extraction technique commonly used in image and signal processing to capture both spatial and frequency information. It offers better frequency localization compared to other wavelet techniques, which is critical in identifying drowsiness-related patterns in physiological signals. Eventually there is a feature selection process upon feature extraction. Feature selection is a process in which it is required to select necessary features that contain a lot of information and remove the features that do not possess that much information. Flemingo which is a feature selection method is incorporated into the process to assists in the selection the best subset features. The selected features are then used as input for Hidden Markov Model (HMM), which is used in identifying more patterns in the text. HMMs are statistical models with application in time series data usually for modelling [27]. In this context, HMM may be used to model the temporal aspect of drowsiness detection since drowsiness of a driver is a process that take place in a given duration of time. Compared to other models, HMMs are capable of modelling changes in drowsiness states as well as making predictions depending on perceived characteristics. HMM's predictions or state transitions are embedded with a classification method. Ranking-based Adaboost integrated regression classifier is referred as well. Adaboost is a kind of ensemble learning which includes a lot of weak classifiers and integrates them into one strong classifier. The second aspect of ranking-based suggests that the classification process may also take into consideration, the confidence of the ranking of the HMM predictions. The classification model uses the Long-Short Term Memory (LSTM) integrated with the AdaBoost classifier for the ranking of the features.

2.2.1 Image pre-processing

The Cross Guided Bilateral Filter is a technique used in image pre-processing for drowsiness detection using the YAWDD and NTHU-DDD datasets. This pre-processing step aims to remove unwanted features and enhance facial regions of interest, allowing images to be subjected to Hidden Markov Models (HMMs) for drowsiness. The Cross Guided Bilateral Filter upgrades the Bilateral Filter, taking into account spatial content and intensity variations while downsampling to eliminate noise while preserving edges and structural components. This helps in analysing drowsiness detection images by preserving certain facial features while eradicating irrelevant ones. The weight of two connected pixels in the original image is calculated in terms of spatial distance, taking into account the Euclidean distance between any two pixels in the original image as estimated from the Eq. (1).

$$W_{Spatial}(p,q) = e^{-\frac{|p-q|^2}{2\sigma_r^2}} \tag{1}$$

"p" and "q" represent the coordinates of two pixels and "σ_r^2" controls the spatial spread of the filter, affecting how much neighboring pixels contribute. The range weight considers the difference in intensity values between two pixels as stated in Eq. (2).

$$W_{range}\,(I(p), I(q)) = e^{-\frac{|I(p)-I(q)|^2}{2\sigma_r^2}} \tag{2}$$

"I(p)" and "I(q)" represent the intensity values of pixels "p" and "q." and "σ_r^2" controls the range spread, determining how different intensity values affect filtering. The Cross-Guidance Weight adjusts the filter based on a guidance image, which is particularly relevant in drowsiness detection to preserve facial features computed with Eq. (3).

$$W_{guidance}(G(p), G(q)) = e^{-\frac{|G(p)-G(q)|^2}{2\sigma_g^2}} \quad (3)$$

"G(p)" and "G(q)" represent the corresponding pixel values in the guidance image and "σ_g^2" controls the guidance spread, influencing the impact of the guidance image. The filtered value of a pixel is computed as a weighted average of nearby pixels, considering spatial, range, and guidance weights computed using Eq. (4).

$$I_{filtered}(p) = \frac{\sum_{q \in N(p)} W_{spatial}(p,q) \cdot W_{range}(I(p), I(q) \cdot W_{guidance}(G(p), G(q)) \cdot I(q)}{\sum_{q \in N(p)} W_{spatial}(p,q) \cdot W_{range}(I(p), I(q) \cdot \mathrm{Wguidance}(G(p), G(q))} \quad (4)$$

In Eq. (4), $I_{filtered}(p)$ represents the filtered value of the pixel at coordinates "p"; $N(p)$ represents a neighborhood of pixels around pixel "p" that are considered in the filtering process; $W_{spatial}(p,q)$ is the spatial Gaussian weight that accounts for the spatial distance between pixels "p" and "q." $W_{range}(I(p), I(q)$ intensity values between pixels "p" and "q." $W_{guidance}(G(p), G(q))$ is the Cross-Guidance weight that adjusts the filter based on a guidance image and I(q) represents the intensity value of pixel "q." This computes the filtered pixel value at location "p" by taking a weighted average of the neighbouring pixel values "q" based on spatial, range, and guidance weights. It effectively enhances the image while preserving important features for drowsiness detection.

Feature extraction with ODT-DL

The process of extracting specific features from images is crucial for distinguishing between drowsy and alert conditions. GLCM captures spatial relations between pixels, describing texture and pattern. Complex Wavelet Transform (CWT) explores details at different resolutions and orientations. Walsh-Hadamard Transform (SWIFT) captures spatial and frequency information, offering better frequency localization for identifying drowsiness-related patterns in physiological signals. This includes SWIFT, GLCM, and DTCWT with Walsh-Hadamard Transform. The wavelet response at a specific orientation and scale can be defined as in Eq. (5).

$$W(x, y; \theta, s) = \int\int I(u,v)\psi\theta, s * (u - x, v - y)dudv \quad (5)$$

In Eq. (5) $W(x, y; \theta, s)$ is the wavelet Response at position (x, y) with orientation θ and scale s. $I(u, v)$ is the input image and $\psi\theta, s$ is the steerable wavelet kernel. The SWIFT algorithm receives a picture, calculates the wavelet responses of the images at different orientations and scales to come up with a feature vector which defines texture of the image. The GLCM matrix is constructed by scanning the image and counting how many times a pixel at position (i, j) takes a specific value. Image Processing based feature extraction evaluates features, the GLCM matrix is used to provide the most information about image texture. In particular, it scans the image for groups of two pixels with a distance of d in a specific direction characterized with θ. For each pixel (x, y)in the image, the GLCM calculates the frequency distribution of the pixel pairs of (i, j) where $I(x, y) = i$ and $I(x + \Delta x, y + \Delta y) = j$ where $(\Delta x, \Delta y)$ is the offset defined by the distance d and the angle θ. In case a pair of pixels lies in this condition, then the particular entry in the GLCM is increased. When the GLCM is computed, other statistics of the image may be obtained in order to quantify the texture of the image. The GLCM features are contrast, Energy, Entropy, Homogeneity and Correlation. The DTCWT coefficients can be computed using a pair of real-valued wavelets: $W_{1,\theta}(x, y) = Re\{I * \psi 1, \theta\}$ and $W_{2,\theta}(x, y) = Im\{I * \psi 1, \theta\}$ in this $W_{1,\theta}(x, y)$ and $W_{2,\theta}(x, y)$ are the real and imaginary components of the DTCWT coefficients at orientation θ. I is the input image. ψ1,θ is the real-valued wavelet at orientation θ. These coefficients capture image structures and texture information in both magnitude and phase.

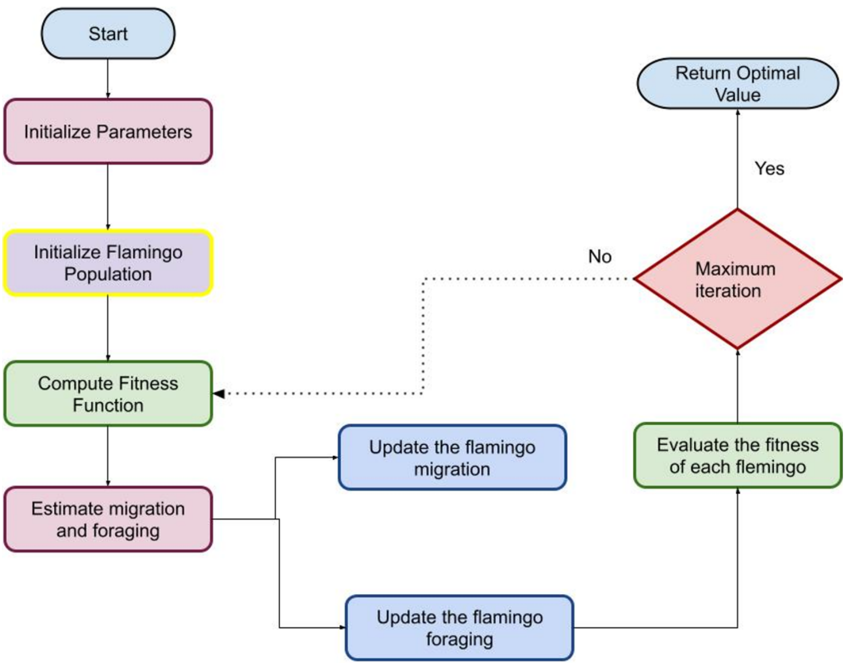


Figure 2. Flow chart of optimization

2.2.2 Feature selection

The process involves extracting a set of features from pre-processed images using methods like SWIFT, GLCM, and Dual-Tree Complex Wavelet Transform with Walsh-Hadamard transform. These features are ranked based on their importance in distinguishing between drowsy and non-drowsy states, represented as R. The Flémingo feature selection algorithm is used to select the most relevant features from F, considering the rankings of individual features. The selected subset of features is denoted as S, where S ⊆ F.

The HMM model integrates selected features S and HMM into a single framework for drowsiness detection. It combines the discriminative power of selected features with HMM's temporal modelling capabilities. Before applying Flémingo, individual features are ranked based on their discriminative power, generating a feature ranking vector (R), with lower ranks being more discriminative. The final subset S is determined based on a threshold or a predetermined number of top-ranked features computed using Eq. (6).

$$S = \{f \in F | rank(f) \leq k\} \tag{6}$$

In Eq. (6), S represents the selected feature subset, F is the set of all features, $rank(f)$ is the rank of feature f, and k is the selected threshold or the desired number of features to be retained. The flow chart of the optimization model with flamingo process is given in Figure 2.

Hidden Markov Model (HMM): The HMM is represented by the following Eqs. (7)-(9).

State Transition Probabilities (A):

$$A(i,j) = P(State_t = j | State_{(t-1)} = i) \tag{7}$$

Emission Probabilities (B):

$$B(j,k) = P(Observation_k | State_t = j) \tag{8}$$

Initial State Probabilities (π):

$$\pi(i) = P(State_1 = i) \tag{9}$$

The integrated ODT-DL model, which combines Flémingo's features with the learned HMM, enhances the accuracy and robustness of detecting driver drowsiness in real-world scenarios. The model calculates the likelihood of observations over time, classifying the driver's state as alert or drowsy based on maximum likelihood estimation or other classification methods. This integration enhances the effectiveness of drowsiness detection in real-world scenarios.

2.2.3 Classification with ODT-DL

Let Ri represent the ranking of the i-th in the training data based on its predicted drowsiness level. Let yi represent the actual drowsiness level of the i-th sample. A ranking loss function, such as the pairwise ranking loss, that quantifies the difference between predicted rankings and actual rankings presented in Eq. (10).

$$\mathrm{L(Ri, Rj, yi, yj) = max(0, \delta - (yi - yi) \cdot (Ri - Rj))} \tag{10}$$

The margin parameter δ controls the degree of ranking violation allowed in the AdaBoost classifier. The features selected by ODT-DL are combined with the ranking-based AdaBoost classifier, which trains the AdaBoost regression model. AdaBoost assigns weights to training samples to emphasize misclassified samples. The integrated AdaBoost regression classifier predicts drowsiness levels based on both features and rankings. The HMM parameters (A, B, π) are estimated using labelled sequences of features corresponding to different drowsiness states. The Baum-Welch algorithm, a variant of the Expectation-Maximization (EM) algorithm, is used for this purpose. The Viterbi algorithm calculates the most likely sequence of states based on observed features.

The forward probabilities $\alpha[i][t]$, which represent the probability of observing the sequence up to time t and being in state I computed as in Eq. (11).

$$\alpha[i][t] = j = 1 \sum N(\alpha[j][t-1] \cdot A[j][i] \cdot B[i][kt]) \tag{11}$$

The backward probabilities β[i][t], which represent the probability of observing the remaining sequence given that you are in state I at time t as stated in Eq. (12):

$$\beta[i][t] = j = 1 \sum N(A[i][j] \cdot B[j][kt+1] \cdot \beta[j][t+1]) \tag{12}$$

The most likely sequence of states (drowsiness levels) by maximizing the joint probability. $q^t = argi\max(\alpha[i][t] \cdot \beta[i][t])$. ODT-DL uses feature extraction and Hidden Markov Models to classify driver drowsiness levels. The Viterbi algorithm calculates the most likely sequence of drowsiness states based on observed features. Key steps within the HMM framework are represented in the given equations.

Algorithm 1. Classification with ODT-DL

```
# Define functions for HMM Forward and Backward algorithms
def forward_algorithm(Observations, A, B, pi):
    T = len(Observations)
    N = len(A)
    alpha = np.zeros((N, T))
      # Initialization
    for I in range(N):
        alpha[i][0] = pi[i] * B[i][Observations[0]]
     # Forward recursion
    for t in range(1, T):
        for j in range(N):
            for I in range(N):
                alpha[j][t] += alpha[i][t-1] * A[i][j]
            alpha[j][t] *= B[j][Observations[t]]
        return alpha
def backward_algorithm(Observations, A, B):
    T = len(Observations)
    N = len(A)
    beta = np.zeros((N, T))
    # Initialization
    for I in range(N):
        beta[i][T-1] = 1.0
      # Backward recursion
    for t in range(T – 2, -1, -1):
        for I in range(N):
            for j in range(N):
                beta[i][t] += A[i][j] * B[j][Observations[t+1]]
* beta[j][t+1]
        return beta
# Define the ODT-DL drowsiness detection algorithm
def CgDTO_HMM_Drowsiness_Detection(TrainingData, TestData):
    # Feature Extraction and Selection using ODT-DL
```

```
SelectedFeatures =
feature_extraction_and_selection(TrainingData)
    # HMM Initialization
N = 2  # Number of states (alert and drowsy)
A = initialize_transition_probabilities(N)
B = initialize_emission_probabilities(N, SelectedFeatures)
pi = initialize_initial_state_probabilities(N)
  # Train HMM using the Baum-Welch algorithm
A, B, pi = train_HMM(TrainingData, N, A, B, pi)
  # Drowsiness Classification for each test sequence
DrowsinessLabels = []
for sequence in TestData:
    # Apply the Forward Algorithm
    alpha = forward_algorithm(sequence, A, B, pi)
    # Apply the Backward Algorithm
    beta = backward_algorithm(sequence, A, B)
    # Calculate the likelihood of the sequence given the HMM
    sequence_likelihood = sum(alpha[:, -1])
    # Classify the sequence based on likelihood
    if sequence_likelihood > threshold:
        DrowsinessLabels.append("Drowsy")
    else:
        DrowsinessLabels.append("Alert")
    return DrowsinessLabels
```

2.2.4 Multi-scale CNN with LSTM for the automated detection

The system uses a Multi-Scale CNN to extract spatial features from input data, denoted as X, which capture important patterns and characteristics from images or driver behavior data. ODT-DL is applied to select a subset of the most relevant features from X, focusing on the most discriminative attributes for drowsiness detection. The LSTM model is initialized with its architecture parameters. LSTM cells have three gates: an input gate (i_t), a forget gate (f_t), and an output gate (o_t). These gates control the flow of information within the cell. Additionally, LSTM cells have a cell state (c_t) and a hidden state (h_t). The hidden stateh_t is the output of the LSTM cell. The input data is organized into sequences, where each sequence corresponds to a period of driver behavior. The feature vector X selected by ODT-DL is used as input at each time step within the sequence. At each time step t in a sequence the Input gate (i_t), and forget gate (f_t) calculated using Eqs. (13) and (14).

$$i_t = sigmoid(W_i * [h(t-1), X_t] + b_i) \quad (13)$$

$$f_t = sigmoid(W_f * [h(t-1), X_t] + b_i) \quad (14)$$

Cell state update based on Eq. (15):

$$C_t = f_t * C(t-1) + i_t * tanh(W_c * [h(t-1), X_t] + b_c) \quad (15)$$

Output gate calculated using Eq. (16):

$$o_t = sigmoid(W_o * [h(t-1), X_t] + b_o) \quad (16)$$

Hidden state update as in Eq. (17):

$$h_t = o_t * tanh(c_t) \quad (17)$$

The learned feature for the sequence is included in the final hidden state h_t after processing of the whole series. For binary classification, the $h_t$will indicate if the driver is tired (1) or not (0). A concluding fully linked layer with a sigmoid activation function may be used to make the classification as described in Eq. (18).

$$y = sigmoid(W\_y * h_t + b_y) \quad (18)$$

The predicted drowsiness level is represented by y, where you can set a threshold to determine alertness or drowsiness. Temporal dependencies in the data are captured by the last hidden state h_t after processing the whole sequence. This representation is for binary classification that decides if the driver is awake or sleepy. A final fully connected layer with weights W_y and a bias term (b_y) followed by a Sigmoid activation function gives us predicted drowsiness level (y). The integrated model is trained using labelled sequences of driver behavior data, optimizing its parameters in order to minimize the binary cross-entropy loss (L) between predicted y and ground truth labels. Evaluation metrics such as accuracy, precision, recall and F1 score are used to evaluate how well this model performs on another testing dataset.

3. EXPERIMENT SETUP

One of the important things to make sure that the results are correct and trustworthy is to set up an experiment for detecting drowsiness by using ODT-DL with datasets such as YAWDD and NTHU-DDD. The arrangement used for drowsiness detection in experimentation has been shown in the Table 1.

Table 1. Experimental setup

Parameters	YAWDD	NTHU – DDD
Training Samples	2000	1500
Testing Samples	500	500
Image Resolution (pixels)	128×128	256×256
Number of Top Features Selected	100	80
LSTM Architecture		
Number of LSTM Layers	2	3
Number of LSTM Units	128	256
Dropout Rate	0.5	0.4
Learning Rate	0.001	0.001
Training Epochs	50	60
Batch Size	32	64
Loss Function	Binary Cross-Entropy	Binary Cross-Entropy
Optimizer	Adam	RMSprop

The categorization process employs a deep learning framework called Long Short-Term Memory (LSTM), trained on specific features, and optimized using an optimizer like Adam or RMSprop for binary cross entropy.

4. RESULTS AND DISCUSSION

The ODT-DL simulation results are an important part of our research on drowsiness detection. This section contains the simulation results and their detailed interpretation, along with a discussion of their applicability to real-world drowsiness detection practices.

Table 2. Performance of ODT-DL

Metric	YAWDD Dataset	NTHU-DDD Dataset
True Positives	1210	990
True Negatives	1360	900
False Positives	40	90
False Negatives	30	20
Accuracy	0.99	0.99
Precision	0.96	0.91
Recall	0.97	0.98
F1-Score	0.97	0.94
ROC AUC	99.5%	99.2%
Kappa	0.9908	0.9929
FPR	2.81%	9.09%
FNR	2.43%	1.98%

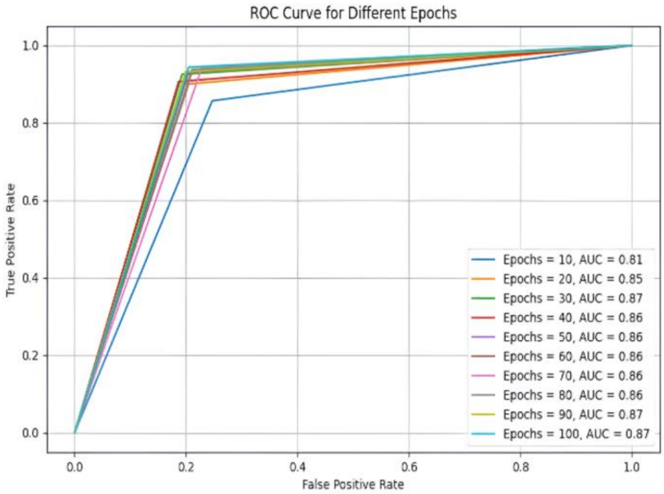


Figure 3. ROC curve for the ODT-DL

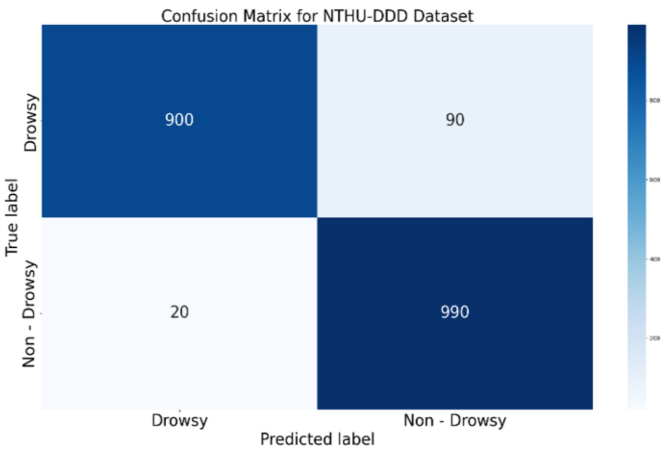


Figure 4. Confusion matrix for NTHU-DDD

The ODT-DL method was tested on two drowsiness detection datasets, YAWDD and NTHU-DDD (Table 2). The model achieved 99% accuracy on the YAWDD dataset, with 1210 true positives and 1360 true negatives, indicating its ability to differentiate between drowsy and non-drowsy situations. Its precision was 96%, and recall was 97%, indicating balanced predictions. On the NTHU-DDD dataset, it achieved 99% accuracy, separating alertness and drowsiness. The ROC AUC scores were 99.2% and 99.5%, confirming the method's reliability against random chances. Despite a low false-positive rate, the false-negative rate was remarkably low at 2.43% and 1.98%. These results confirm the effectiveness of the ODT-DL approach in drowsiness detection, highlighting its potential for real-world applications in ensuring road safety and driver alertness. The Figure 3 illustrated the ROC curve computed for the proposed ODT-DL model for the estimation of the features. The proposed ODT-DL performance for the classification is computed and confusion matrix are presented in Figure 4 and Figure 5.

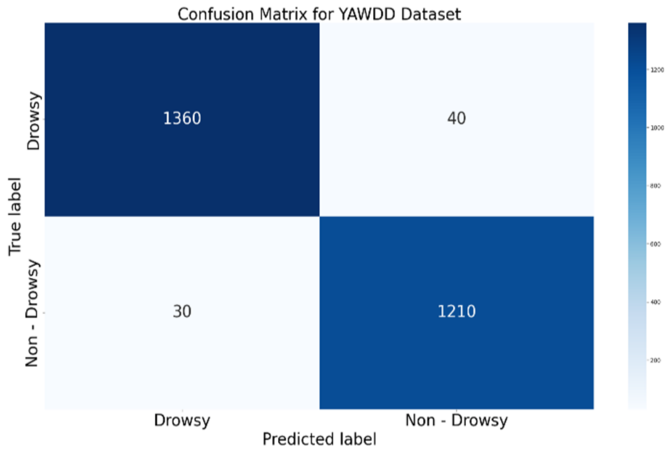


Figure 5. Confusion matrix for YAWDD

Table 3 presents a comparative analysis of machine learning and deep learning models for detecting drowsiness using the YAWDD dataset. The ODT-DL model is the most effective, achieving an accuracy of 99% in classifying drivers as either drowsy or alert in 99% of cases, indicating its superior performance across critical metrics. Other important measures in this field include precision and recall, which ODT-DL scores highly on. It means that the model has an appropriate blend of identifying sleepy drivers accurately (recall) and avoiding unnecessary alerts (precision) at 96% and 97% respectively. The F1-score, which combines precision and recall, is 97% robust in detecting drowsiness, while the ROC AUC (Receiver Operating Characteristic Area Under Curve) score, valued at 99.5%, distinguishes between sleepy and non-sleepy states, making it a significant measure. This indicates that there is a clear contrast between these two states making it more applicable. Similarly, its low false positive rate (FPR) is 2.81% while its False Negative Rate (FNR) amounts to 2.43% which show that detecting sleepy drivers with minimal alarm false alarm production are possible. Compared with several models such as AlexNet, ResNet, VggNet, SVM, Random Forest, KNN, NaiveBayes and AdaBoost; however, ODT-DL model outshines them all when it comes to precision and general performance but among them their accuracies differ.

With the YAWDD dataset the comparative analysis is presented in Figure 6(a) - Figure 6(l).

Table 4 and Figure 7(a) - Figure 7(l) presents a comprehensive comparative analysis of various machine learning and deep learning models for the detection of drowsiness using the NTHU-DDD dataset. The analysis of various models for detecting drowsiness shows that the Cross Guided Dual-Tree Optimization – Hidden Markov Model (ODT-DL) is the top-performing model. It achieves an accuracy rate of 99%, indicating its ability to differentiate between drowsy and alert drivers in 99% instances. This precision is crucial for enhancing road safety. The model also performs well in precision and recall, with a precision score of 91% and recall of 98%. Its high F1-Score of 94% indicates its robustness in detecting driver drowsiness. The Receiver Operating Characteristic Area Under Curve (ROC AUC) ranges between 0.91 and 0.99, confirming its excellent discriminatory power. The model also shows minimal false alarms, with FPR and FNR of 9.09% and 1.98% respectively, indicating its effectiveness in detecting drowsy drivers while minimizing false alarms. Other models like AlexNet, ResNet, VggNet, SVM, Random Forest, KNN, NaiveBayes, and AdaBoost do not perform well.

Table 3. Comparative analysis for the YAWDD dataset

Metric	ODT-DL (YAWDD)	AlexNet	ResNet	VggNet	SVM	Random Forest	KNN	NaiveBayes	AdaBoost
True Positives	1210	1180	1010	1175	1050	1120	990	980	1055
True Negatives	1360	1340	920	1350	880	1335	910	890	930
False Positives	40	60	120	55	80	65	110	120	95
False Negatives	30	50	20	45	30	40	20	30	25
Accuracy	0.99	0.97	0.96	0.98	0.95	0.97	0.96	0.95	0.97
Precision	0.96	0.95	0.89	0.96	0.91	0.94	0.90	0.88	0.92
Recall	0.97	0.96	0.98	0.97	0.95	0.97	0.98	0.97	0.96
F1-Score	0.97	0.95	0.92	0.96	0.93	0.95	0.92	0.92	0.94
ROC AUC	99.5%	98.7%	98.5%	98.9%	97.8%	98.6%	98.4%	98.2%	98.8%
Kappa	0.9908	0.9705	0.9573	0.9721	0.9372	0.9635	0.9521	0.9301	0.9654
FPR	2.81%	4.48%	8.54%	3.93%	7.28%	4.67%	8.47%	9.86%	6.42%
FNR	2.43%	4.20%	1.90%	3.85%	2.64%	2.68%	1.91%	2.34%	2.36%

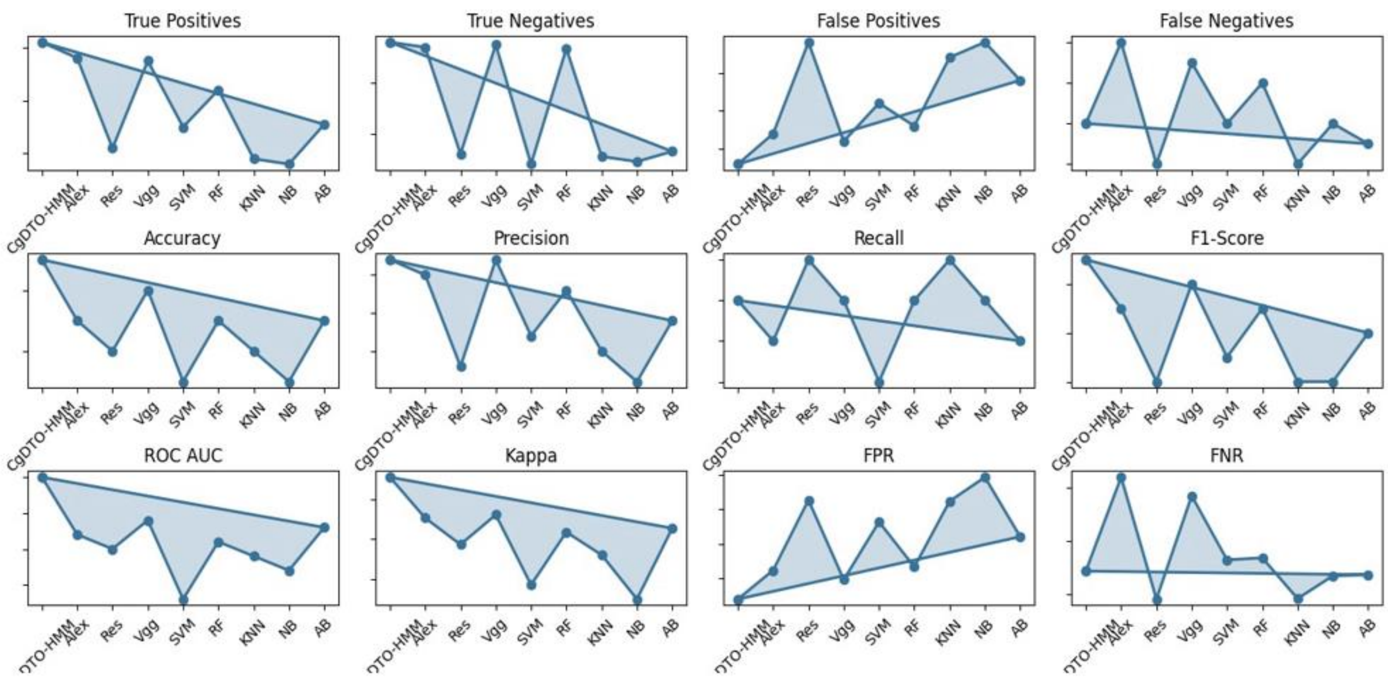


Figure 6. Comparison of ODT-DL for the YAWDD dataset

Table 4. Comparative analysis for the NTHU-DDD

Metric	ODT-DL (NTHU-DDD)	AlexNet	ResNet	VggNet	SVM	Random Forest	KNN	NaiveBayes	AdaBoost
True Positives	990	930	950	980	960	980	910	950	1000
True Negatives	900	920	940	910	880	920	940	890	930
False Positives	90	120	60	90	80	60	120	110	80
False Negatives	20	50	20	30	30	20	50	30	20
Accuracy	0.99	0.97	0.96	0.98	0.95	0.97	0.96	0.95	0.97
Precision	0.91	0.89	0.94	0.92	0.91	0.94	0.88	0.89	0.93
Recall	0.98	0.98	0.95	0.97	0.95	0.98	0.95	0.97	0.98
F1-Score	0.94	0.92	0.94	0.94	0.93	0.95	0.91	0.93	0.95
ROC AUC	99.2%	98.5%	98.2%	98.6%	97.4%	98.3%	98.1%	97.9%	98.4%
Kappa	0.9929	0.9712	0.9584	0.9723	0.9374	0.9631	0.9517	0.9304	0.9657
FPR	9.09%	7.18%	3.85%	6.25%	8.23%	4.35%	7.75%	8.96%	5.71%
FNR	1.98%	1.98%	1.98%	2.98%	2.98%	1.98%	2.98%	2.98%	1.98%

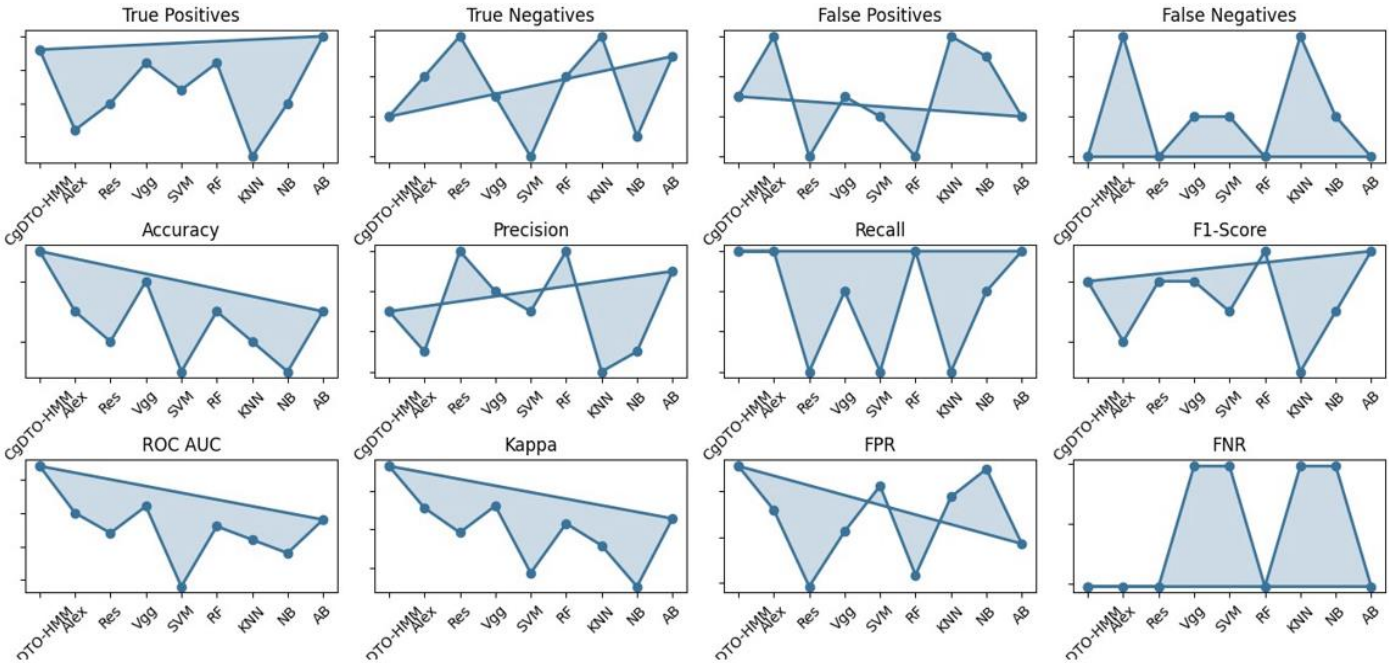
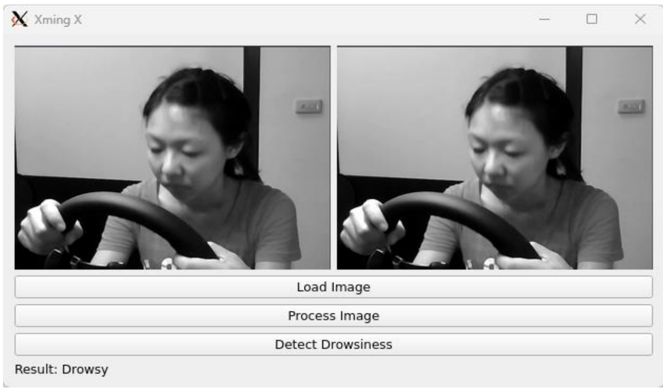


Figure 7. Comparison of ODT-DL for the NTHU-DDD dataset



(a)

(b)

Figure 8. Drowsiness detection with ODT-DL GUI (a) Drowsy (b) Normal

With the proposed ODT-DL model creates a GUI with drowsy or alert images based on simplified analysis shown in Figure 8(a) and Figure 8(b). The algorithm measures average pixel intensity in black and white images, classifying them as "Drowsy" if below a threshold and "Normal" if exceeding it. This model outperforms other models in the YAWDD and NTHU-DDD databases with high accuracy levels above 99%.

It also yields optimal performance with high precision, recall, and F1 Score, allowing it to differentiate between drowsy and non-drowsy states with low false alarms. This is crucial in real-life situations where classifying a driver as sleepy when they are actually not can lead to disastrous consequences. Other models like AlexNet, ResNet, and SVM also achieved good results but never surpassed the results achieved by ODT-DL. The ROC AUCs and Kappa coefficients of ODT-DL are high, indicating the model's ability to distinguish between drowsy and non-drowsy states. The low figures in false positive and false negative cases also indicate low chances of producing unnecessary alarms while identifying drowsy drivers. The results confirm the potential of sophisticated machine learning algorithms like ODT-DL to improve road safety by providing a reliable and accurate tool for monitoring driver drowsiness. Future studies and practical application of such models in vehicles' safety systems could help avoid accidents and save lives on the road.

The ODT-DL machine learning model has demonstrated high effectiveness in detecting drowsiness in drivers, with an accuracy of over 99% in both YAWDD and NTHU-DDD databases. The model demonstrated high precision, recall, and F1-Score, indicating its ability to avoid false detection, making it crucial for practical applications, enabling accurate detection of drugged drivers without additional stress. The ODT-DL model outperforms other machine learning models like SVM and AlexNet ResNet in distinguishing between drowsy and non-drowsy states with higher ROC AUC values. The ODT-DL model enhances road safety and drowsiness detection through improved prediction and actual classification, minimizing false positives and negatives, indicating its high accuracy and potential for further development. Future developments and applications of these models could significantly impact accident risk reduction and life preservation on the roads.

5. CONCLUSION

The ODT-DL mathematical model has demonstrated exceptional effectiveness in detecting drowsy drivers. Experiments on the YAWDD and NTHU-DDD databases showed that ODT-DL can achieve an accuracy of over 99%, making it a reliable method for distinguishing between alert and drowsy states. This high accuracy is crucial in real-life applications, where identifying drowsiness at the right time can eliminate accidents. ODT-DL also achieved high precision, recall, and F1-Score, reducing false alarms and correcting drowsy drivers. The ROC AUC values are high, indicating high discrimination ability. The Kappa coefficient indicates that predicted classifications match reality at a fast rate. Compared to other machine learning models, ODT-DL performed competitively but lagged behind. ODT-DL has minimal false positives and false negatives, making it more practical for implementation in vehicular safety systems. The findings suggest that sophisticated models like ODT-DL can predict drivers' drowsiness, enhancing road safety. Future use in driving and vehicle safety research and implementation in vehicle safety systems could further reduce accidents and improve road safety.

REFERENCES

[1] Zhu, T., Zhang, C., Wu, T., Ouyang, Z., et al. (2022). Research on a real-time driver fatigue detection algorithm based on facial video sequences. Applied Sciences, 12(4): 2224. https://doi.org/10.3390/app12042224

[2] Bekhouche, S.E., Ruichek, Y., Dornaika, F. (2022). Driver drowsiness detection in video sequences using hybrid selection of deep features. Knowledge-Based Systems, 252: 109436. https://doi.org/10.1016/j.knosys.2022.109436

[3] Jahan, I., Uddin, K.A., Murad, S.A., Miah, M.S.U., et al. (2023). 4D: A real-time driver drowsiness detector using deep learning. Electronics, 12(1): 235. https://doi.org/10.3390/electronics12010235

[4] Vijaypriya, V., Uma, M. (2023). Facial feature-based drowsiness detection with multi-scale convolutional neural network. IEEE Access, 11: 63417-63429. https://doi.org/10.1109/ACCESS.2023.3288008

[5] Walizad, M.E., Hurroo, M., Sethia, D. (2022). Driver drowsiness detection system using convolutional neural network. In 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, pp. 1073-1080. https://doi.org/10.1109/ICOEI53556.2022.9777182

[6] Chen, J., Yan, M., Zhu, F., Xu, J., Li, H., Sun, X. (2022). Fatigue driving detection method based on combination of BP neural network and time cumulative effect. Sensors, 22(13): 4717. https://doi.org/10.3390/s22134717

[7] Al Redhaei, A., Albadawi, Y., Mohamed, S., Alnoman, A. (2022). Realtime driver drowsiness detection using machine learning. In 2022 Advances in Science and Engineering Technology International Conferences (ASET), Dubai, United Arab Emirates, pp. 1-6. https://doi.org/10.1109/ASET53988.2022.9734801

[8] Faisal, T., Negassi, I., Goitom, G., Yassin, M., Bashir, A., Awawdeh, M. (2022). Systematic development of real-time driver drowsiness detection system using deep learning. IAES International Journal of Artificial Intelligence, 11(1): 148. https://doi.org/10.11591/ijai.v11.i1.pp148-160

[9] Zhao, G., He, Y., Yang, H., Tao, Y. (2022). Research on fatigue detection based on visual features. IET Image Processing, 16(4): 1044-1053. https://doi.org/10.1049/ipr2.12207

[10] Husain, S.S., Mir, J., Anwar, S.M., Rafique, W., Ullah, M.O. (2022). Development and validation of a deep learning-based algorithm for drowsiness detection in facial photographs. Multimedia Tools and Applications, 81(15): 20425-20441. https://doi.org/10.1007/s11042-022-12433-x

[11] Zu, S., Jin, Y., Yang, D., Xu, H. (2022). DrowsyNet: Multivariate time series classification for embedded driver drowsiness detection. In 2022 8th International Conference on Control, Automation and Robotics (ICCAR), Xiamen, China, pp. 442-451. https://doi.org/10.1109/ICCAR55106.2022.9782626

[12] Xiao, W., Liu, H., Ma, Z., Chen, W., Sun, C., Shi, B. (2022). Fatigue driving recognition method based on multi-scale facial landmark detector. Electronics, 11(24): 4103. https://doi.org/10.3390/electronics11244103

[13] Zaman, M., Saha, S., Puryear, N., Zohrabi, N., Abdelwahed, S. (2022). Incorporation of physiological features in drowsiness detection using deep neural network approach. In 2022 IEEE Transportation Electrification Conference & Expo (ITEC), Anaheim, CA, USA, pp. 219-224. https://doi.org/10.1109/ITEC53557.2022.9813946

[14] Albadawi, Y., AlRedhaei, A., Takruri, M. (2023). Real-time machine learning-based driver drowsiness detection using visual features. Journal of imaging, 9(5): 91. https://doi.org/10.3390/jimaging9050091

[15] Siwach, M., Mann, S., Gupta, D. (2022). A practical implementation of driver drowsiness detection using facial landmarks. In 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, pp. 1-4. https://doi.org/10.1109/ICRITO56286.2022.9964990

[16] Zhao, C., Gao, Z., Wang, Q., Xiao, K., Mo, Z., Deen, M. J. (2023). FedSup: A communication-efficient federated learning fatigue driving behaviors supervision approach. Future Generation Computer Systems, 138: 52-60. https://doi.org/10.1016/j.future.2022.08.009

[17] Li, D., Zhang, X., Liu, X., Ma, Z., Zhang, B. (2023). Driver fatigue detection based on comprehensive facial features and gated recurrent unit. Journal of Real-Time Image Processing, 20(2): 19. https://doi.org/10.1007/s11554-023-01260-4

[18] Kim, M., Koo, J. (2022). Embedded system performance analysis for implementing a portable drowsiness detection system for drivers. Technologies, 11(1): 8. https://doi.org/10.3390/technologies11010008

[19] Mohamed, G.M., Patel, S.S., Naicker, N. (2023). Data augmentation for deep learning algorithms that perform driver drowsiness detection. International Journal of Advanced Computer Science and Applications, 14(1): 233-248. https://doi.org/10.14569/IJACSA.2023.0140127

[20] Sun, Z., Miao, Y., Jeon, J.Y., Kong, Y., Park, G. (2023). Facial feature fusion convolutional neural network for driver fatigue detection. Engineering Applications of Artificial Intelligence, 126: 106981. https://doi.org/10.1016/j.engappai.2023.106981

[21] Ahmed, M.I.B., Alabdulkarem, H., Alomair, F., Aldossary, D., et al. (2023). A deep-learning approach to

driver drowsiness detection. Safety, 9(3): 65. https://doi.org/10.3390/safety9030065
[22] Ziryawulawo, A., Kirabo, M., Mwikirize, C., Serugunda, J., Mugume, E., Miyingo, S.P. (2023). Machine learning based driver monitoring system: A case study for the Kayoola EVS. SAIEE Africa Research Journal, 114(2): 40-48. https://doi.org/10.23919/SAIEE.2023.10071976
[23] Chang, W., Yang, X., Belhe, Y., Ramamoorthi, R., Li, T.M. (2024). Spatiotemporal bilateral gradient filtering for inverse rendering. In SIGGRAPH Asia 2024 Conference Papers, Tokyo, Japan, pp. 1-11. https://doi.org/10.1145/3680528.3687606
[24] Chen, M.L., Yu, X.H., Su, S.J., Liu, D.Q. (2023). Image texture analysis based on gray level co-occurrence matrix. In 2023 13th International Conference on Information Technology in Medicine and Education (ITME), Wuyishan, China, pp. 451-455. https://doi.org/10.1109/ITME60234.2023.00096
[25] Roodbary, A.A., Bastani, M.H., Behnia, F. (2024). Classification of automotive radar targets using gray level co-occurrence matrix (GLCM). In 2024 32nd International Conference on Electrical Engineering (ICEE), Tehran, Iran, Islamic Republic of, pp. 1-5. https://doi.org/10.1109/ICEE63041.2024.10667870
[26] Teng, H., Jiao, J., Shang, X., Wang, Y., et al. (2023). Two-dimensional complex wavelet transform for linear noise attenuation and image decomposition. Journal of Geophysics and Engineering, 20(3): 474-482. https://doi.org/10.1109/TIV.2021.3065933
[27] Deng, Q., Söffker, D. (2021). A review of HMM-based approaches of driving behaviors recognition and prediction. IEEE Transactions on Intelligent Vehicles, 7(1): 21-31. https://doi.org/10.1109/TIV.2021.3065933

NOMENCLATURE

S	Represents the selected feature subset
F	Set of all features
k	Selected threshold or the desired number of features to be retained
Ri	Ranking of the i-th in the training data
X	Extract spatial features from input data
i_t	Input gate
f_t	Forget gate
o_t	Output gate
c_t	Cell state
h_t	Hidden state
W_y	Weights
b_y	Bias term
y	Predicted drowsiness level
L	Binary cross-entropy loss
$rank(f)$	Rank of feature f
(A, B, π)	HMM (Hidden Markov Model) parameters
$\alpha[i][t]$	Probability of observing sequence up to time *t* in state *i*
$\beta[i][t]$	Probability of observing remaining sequence given state *i* at *t*

Greek symbols

σ_r^2	Controls the spatial spread of the filter
σ_g^2	Controls the guidance spread
$\psi\theta$	Steerable wavelet kernel
$(\Delta x,\Delta y)$	Offset
$\psi 1,\theta$	Real-valued wavelet at orientation θ
$S \subseteq F$	Denote the selected subset of features as S
δ	Margin parameter
α	AdaBoost assigns weights

Subscripts

p , q	Coordinates of two pixels
$I(p), I(q)$	Intensity values of pixels "p" and "q."
$G(p), G(q)$	Corresponding pixel values in the guidance image
$I_{filtered}(p)$	Filtered value of the pixel at coordinates "p"
$W_{spatial}(p,q)$	Spatial Gaussian weight
W_{range} $(I(p), I(q))$	Range Gaussian weight
$W_{guidance}$ $(G(p), G(q))$	Cross-Guidance weight using guidance image values
$W(x, y; \theta, s)$	Wavelet response at position (x, y) with orientation θ and scale s
$I(u, v)$	Input image
$W_{1,\theta}(x, y)$ and $W_{2,\theta}(x, y)$	Real and imaginary components of wavelet response