ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

DEEP LEARNING FOR MULTI-CLASS SLEEPINESS CLASSIFICATION WITH MLP AND SMOTE-ENHANCED DATA BALANCING BASED ON EYE ASPECT RATIO

Ahmad Zaini[1], Eko Mulyanto Yuniarno[2], Yoyon K Suprapto[3], Annida Miftakhul Farodisa[4]

[1]Electrical Engineering, Institut Teknologi Sepuluh Nopember
[1,2,3,4]Computer Engineering, Institut Teknologi Sepuluh Nopember

Email: zaini@its.ac.id[1], ekomulyanto@its.ac.id[2], yoyonsuprapto@ee.its.ac.id[3], annidamf@gmail.com[4]

Abstract

This paper addresses the challenge of accurately classifying sleepiness levels based on the Karolinska Sleepiness Scale (KSS) using Eye Aspect Ratio (EAR) data, especially when class imbalance leads to biased predictions. The research proposes a deep learning framework that integrates a Multi-Layer Perceptron (MLP) with the Synthetic Minority Over-sampling Technique (SMOTE) to balance the training data. EAR features, representing eye closure patterns, are extracted from video frames, and SMOTE is applied to generate synthetic data for underrepresented sleepiness classes. By training the MLP model on this balanced dataset, the system achieves a 97.6% classification accuracy in distinguishing four distinct sleepiness levels based on the KSS, demonstrating its effectiveness in reducing prediction bias and managing class imbalance, both crucial for real-time drowsiness detection systems.

Keywords: Classification, Deep Learning, Eye Aspect Ratio, Karolinska Sleepiness Scale, Multi-Layer Perceptron, Sleepiness Scale

Received: 12-09-2024 | **Revised:** 21-10-2024 | **Accepted:** 01-11-2024
DOI: https://doi.org/10.23887/janapati.v13i3.84962

INTRODUCTION

Driving requires full psychological and cognitive alertness to ensure driver performance. Lapses in these areas can have fatal outcomes, such as accidents. According to the American Automobile Association (AAA) Foundation for Traffic Safety's 2019 Traffic Safety Culture Index, common driving issues include distraction, aggressive driving behavior, drowsiness, and psychological and physical impairments. Among respondents, 96% indicated that drowsy driving is extremely dangerous; however, 24% admitted to driving while so tired it was difficult to keep their eyes open at least once [1]. Drowsiness is a state between wakefulness and sleep, detectable through brain activity (reflecting information processing capacity) and eye activity (indicating visual perception).

A drowsiness detection system aims to prevent accidents caused by driver fatigue. One intrusive method for detecting sleepiness involves extracting Heart Rate Variability (HRV) data, which reflects autonomic nervous system activity. Vicente et al. studied 17 men and 13 women in a driving simulation, calculating HRV using an Electroencephalogram (EEG) for brain activity and an Electrocardiogram (ECG) for heart activity. The results indicated that as drowsiness increases, heart rate decreases [2].

Non-intrusive methods in drowsiness detection utilize visual data processed by machine learning and deep learning models to identify drowsy facial expressions [3] and extreme head positions [4]. These approaches generally rely on either the full frontal facial image for expression classification or specific facial features for more detailed analysis [5]. Local features like the eyes and mouth often indicate drowsiness [6]. Observable signs, such as prolonged eye closure (PERCLOS) [7-8], frequent blinking [9], and yawning [10], are monitored through cameras, allowing continuous facial observation without interrupting the user's activity. Convolutional Neural Networks (CNN) analyze this data in real-time, effectively detecting signs of drowsiness and providing early warnings to drivers.

Fauzia et al. developed a visual-based drowsiness detection system that tracks eye blinks. If prolonged eye closure is detected, the system monitors eye movement and triggers an alarm through a vibrator. Built with OpenCV and a Raspberry Pi camera, experiments demonstrate that this system effectively detects drowsiness and can help reduce vehicle accidents [11].

Using the Eye Aspect Ratio (EAR) feature, Maior et al. (2020) developed a non-intrusive sleep detection method using a simple web camera. The model, tested for various eye behaviors via machine learning and computer vision, achieved an average test accuracy of 94.9%. Validated with the DROZY dataset, the model's accuracy was further supported using the Karolinska Sleepiness Scale (KSS) and response time as indicators of driver performance [12]. The Karolinska Sleepiness Scale, developed by researchers at the Karolinska Institute in Sweden, is a subjective tool widely used to evaluate individual levels of sleepiness or alertness, especially in studies on fatigue, alertness, and sleep deprivation [13].

Developing a non-intrusive sleepiness classification model presents several challenges. Existing classifications, limited to two [11], [14] or three [15] categories, restrict the model's ability to accurately represent nuanced sleepiness levels. Limited frame data hindered the model's representation of continuous eye-opening and -closing patterns. As a result, the model underrepresented complex sleep-induced eye movements. Furthermore, the interpolation method used to address missing data was suboptimal due to limited data, and unbalanced class data introduced bias, leading to inaccurate sleepiness classifications. This study aims to create a framework for classifying sleepiness into four classes based on EAR, aligning with the Karolinska Sleepiness Scale. Addressing data imbalance within each class, this framework seeks to provide balanced training data, reducing potential bias during training.

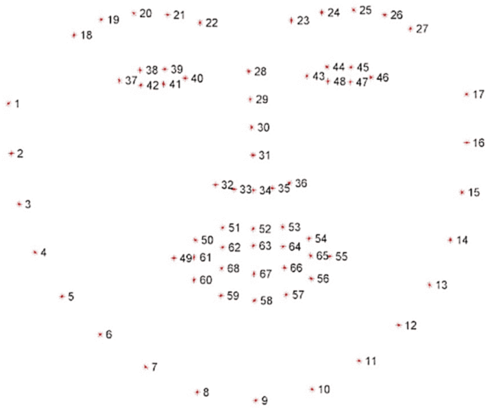


Figure 1. The 68 facial landmarks configuration [18]

RELATED WORKS

A. Karolinska Sleepiness Scale (KSS)

The Karolinska Sleepiness Scale (KSS) is a subjective sleepiness assessment tool, allowing subjects to select statements that best describe their current state. KSS scores range from 1 (very alert) to 9 (very sleepy), as shown in Table *1* [12]. KSS is widely applied to assess sleepiness in shift workers, drivers, jetlag sufferers, and patients with specific medical conditions. Research shows a high correlation between KSS scores, EEG variables, and human behavior, indicating high validity for the KSS as a measure of sleepiness [13].

Table 1. KSS Description

Rate	Verbal description
1	Extremely alert
2	Very alert
3	Alert
4	Fairly alert
5	Neither alert nor sleepy
6	Some signs of sleepiness
7	Sleepy but no effort to stay awake
8	Sleepy, some effort to stay awake
9	Very sleepy, great effort to stay awake, fighting sleep

B. Face Landmarks

Face landmarks are sets of points representing key facial features, as illustrated in Figure *1*. They are essential in facial image processing applications, including head position estimation, emotion classification, face alignment, and facial recognition. Various algorithms detect the location of critical facial feature points, such as the eye corners [16]. In this research, face landmarks are used to detect the face and eyes in each frame, enabling calculation of the EAR value. One notable method for implementing face landmarks, presented by Kazemi et al. (2014) [17], employs HOG (Histogram of Oriented Gradients) for face detection and a Linear SVM classifier, producing 68 coordinate points representing the face, accessible via the Dlib module [18]**Error! Reference source not found.**.

C. Eye Aspect Ratio (EAR)

The Eye Aspect Ratio (EAR) is a measure based on the ratio of eye length to width, serving as an indicator of eye openness [19]. Six reference points around the eye are used to calculate EAR from the vertical and horizontal distances between these points. In **Error! Reference source not found.**, points $p_1 - p_6$

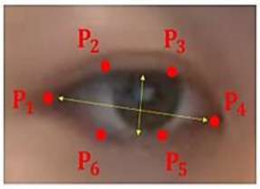


Figure 2. Six landmark points. Adapted from [20]

represent the eyes. The horizontal line between p_1 and p_4 denotes eye length, while the vertical line, derived from the midpoint between $p_2 - p_3$ and $p_5 - p_6$, represents eye width or the vertical Euclidean distance. The segment between p_1 and p_4 represents the horizontal Euclidean distance for eye opening. Eq. 1 represents the formula for calculating the Eye Aspect Ratio (EAR):

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|} \qquad (1)$$

The EAR value tends to be higher and more stable when the eyes are open, whereas it approaches zero when the eyes are closed. This value can be utilized in computer vision to determine eye status (open or closed) [12].

METHODOLOGY

This study proposes a framework to develop a four-class sleepiness classification model, based on EAR values and aligned with the KSS standard. Using deep learning and data augmentation techniques, SMOTE is applied to balance data across classes during training. The primary contribution is a non-intrusive sleepiness classification model capable of identifying four levels of sleepiness based on the KSS. By leveraging EAR as the main indicator, this study integrates a Multi-Layer Perceptron (MLP) with SMOTE to address data imbalance, thereby enhancing model accuracy and generalizability. Figure *3* illustrates the research methodology.

A. DROZY Dataset's NIR Videos

The DROZY dataset is used as the ground truth for model training and testing. Developed by the Laboratory for Signal and Image Exploitation (INTELSIG) at the University of Liège (ULg) in Belgium, this dataset includes Near Infrared (NIR) video recordings of 14 European/Caucasian subjects who each completed a psychomotor vigilance test (PVT) three times while driving. Subjects were instructed to stay awake for one day, five hours, and thirty minutes prior to testing, as shown in Figure *4*. Before beginning each PVT session, subjects recorded their KSS score to represent their sleepiness level [21].

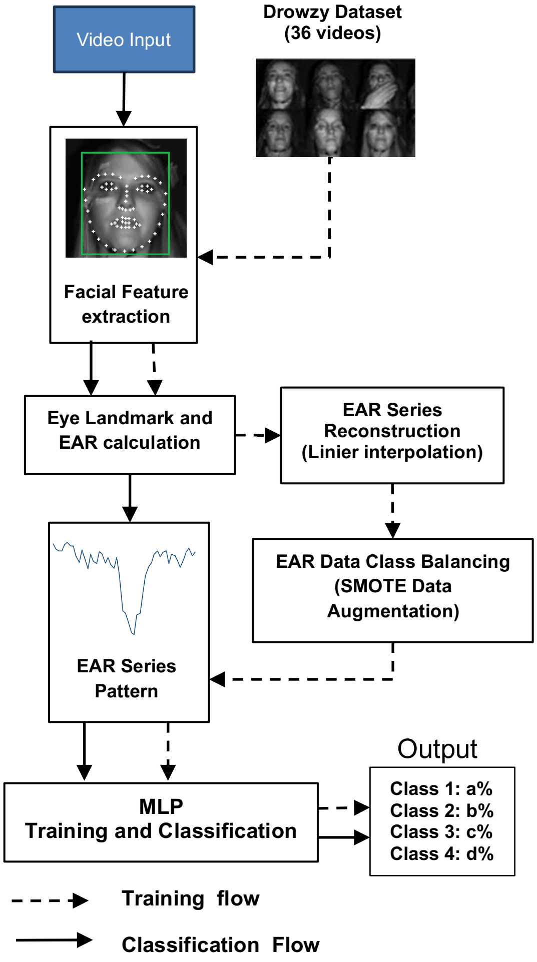
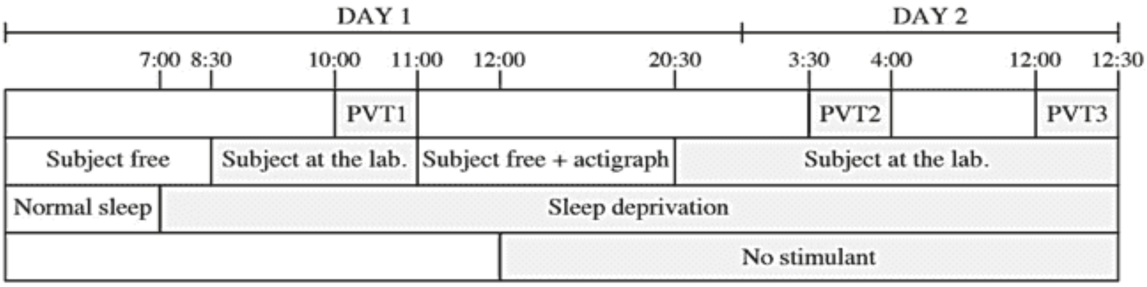

Video Input
Drowzy Dataset
(36 videos)
Facial Feature extraction
Eye Landmark and EAR calculation
EAR Series Reconstruction (Linier interpolation)
EAR Data Class Balancing (SMOTE Data Augmentation)
EAR Series Pattern
Output
MLP Training and Classification
Class 1: a%
Class 2: b%
Class 3: c%
Class 4: d%
Training flow
Classification Flow

Figure 3. Research Methodology

KSS scores, initially ranging from one to nine, were grouped into four classes based on the similarity of conditions represented by the Karolinska Sleepiness Scale. This approach is adapted from Ghoddoosian et al. [22], who grouped nine KSS scores into three classes, excluding scales 4 and 5.



1) Class 1: KSS 2 and 3 (alert) – The subject is alert with no signs of sleepiness.
2) Class 2: KSS 4 and 5 (low vigilance) – The subject is in a state between alertness and sleepiness.
3) Class 3: KSS 6 and 7 (sleepy) – The subject is drowsy but can attempt to stay awake.
4) Class 4: KSS 8 and 9 (very drowsy) – The subject is extremely sleepy, struggling to stay awake.

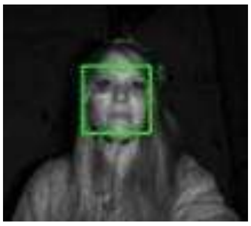


Figure 5. Face detection in one frame

B. Eye Feature Extraction

Using a pretrained face detector and landmark detection based on HOG and Linear SVM in the Dlib module [17], facial features were extracted to obtain face landmarks. At this stage, eye feature extraction was performed to identify six points on the eye landmarks. Figure *5* displays the region of interest (ROI) from face detection on a single frame, while Figure *7* shows the 68 facial landmark points detected in each frame.

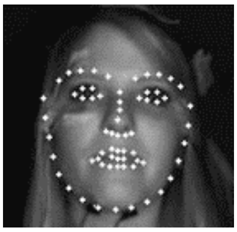


Figure 7. Face landmark detection results on one frame

To calculate the EAR value, detection focused on eye features (reference points 36–41 and 42–47). Figure *8* illustrates the open eye state with six landmark points. For each eye, six (x, y) coordinates were identified, allowing for the calculation of the EAR value per eye based on Eq. 1. The average EAR value was determined by averaging the EARs of both eyes. When the eyes were open, the EAR value was 0.3 or higher, while it was less than 0.3 and approached zero when the eyes were closed. **Error! Reference source not found.** presents an EAR graph illustrating eye states for open and closed conditions, showing a stable EAR value when the eyes are open and a sharp decrease when they are closed.

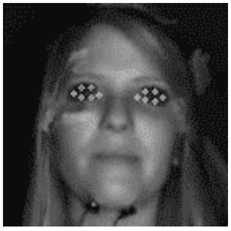


Figure 8. Open eye state landmark points

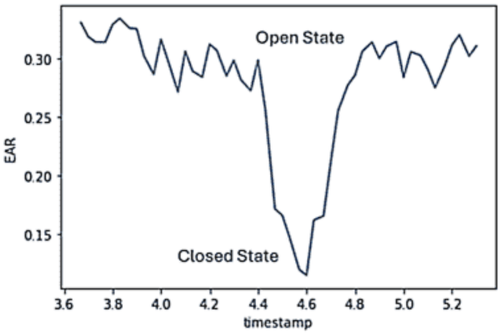


Figure 6. Eye state illustrated by EAR values

C. EAR Series Data Reconstruction

Variations in framerate result in a different number of frames for each video of the same duration. Additionally, video recording limitations cause some frames to be lost [21]. Data loss also occurs when poor face detection leads to Spontaneous subject movements that obscure the face can prevent facial feature detection, resulting in further frame data loss. This data loss can introduce bias between recorded results and subjective sleepiness scale scores.

To address this issue, linear interpolation is used to estimate the eye landmark coordinates for the missing frames. The goal is to reconstruct landmark data across approximately 18,000 frames per video (assuming 30 fps for a 10-minute video) to avoid data asynchrony. Since the data is in a two-dimensional space (x, y) and most missing data occurs between sequential frame indices, linear interpolation is applied. The interpolation follows Eq. 2:

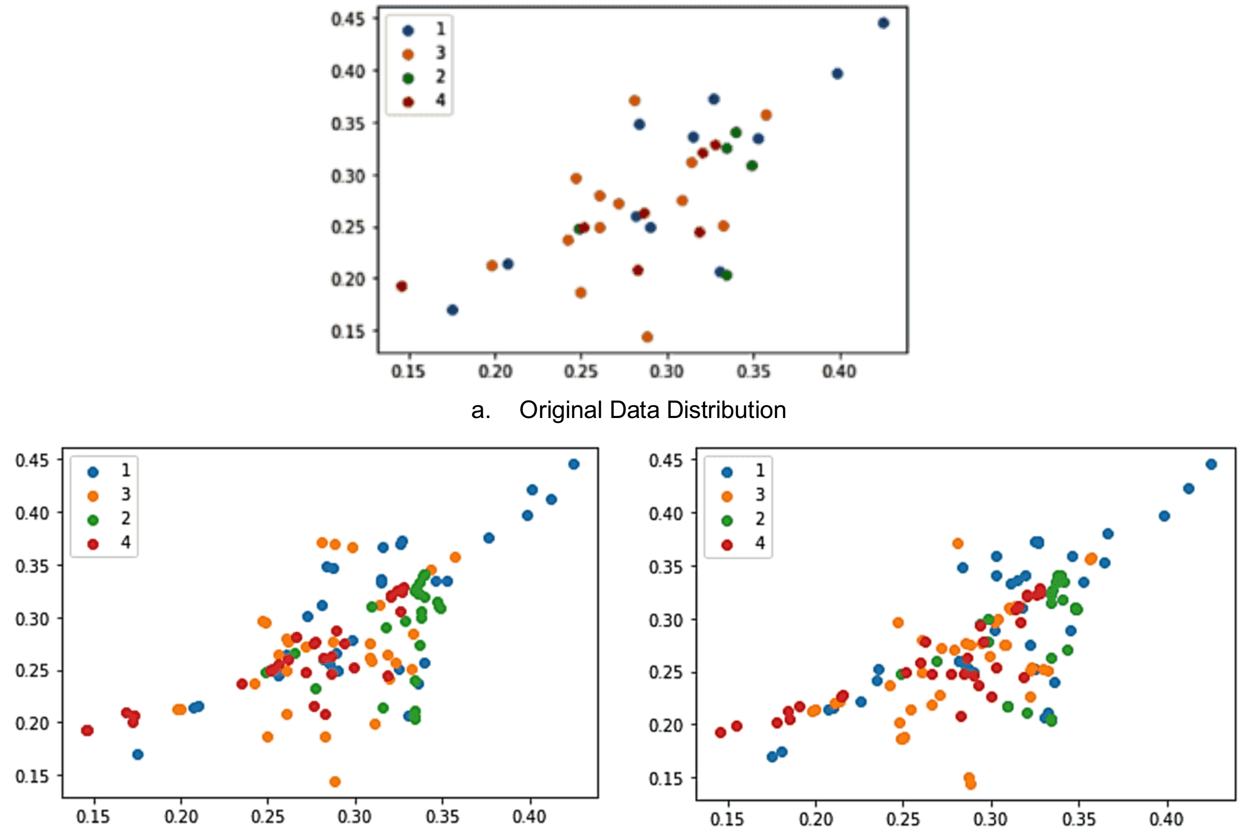
$$y = y_1 + \left(\frac{y_2 - y_2}{x_2 - x_1}\right)(x - x_1) \tag{2}$$

Here, (x, y) represents the interpolated value between points (x_1, y_1) and (x_2, y_2). In this study, x represents the frame index, and y represents the eye landmark coordinate.

D. EAR Data Class Balancing

The initial data exhibits an unbalanced class distribution, which can negatively impact classification performance. To address this, oversampling is conducted using the SMOTE technique, which generates synthetic data in the specified proportions [23]. SMOTE is selected for its ability to create synthetic data that closely resembles the original without duplication [23], [24]. This approach is implemented using the *imblearn* module, with several key parameters influencing the generated synthetic data:

1) Sampling strategy: Defines the oversampling approach, which can specify the percentage of oversampling, target specific classes (minority, majority, or all), or set a desired data count per class.



a. Original Data Distribution

b. SMOTE Data Augmentation Scenario A

c. SMOTE Data Augmentation Scenario B

Figure 9. Comparison of Data Distribution between Original Data and Post-SMOTE Oversampling in Scenarios A and B

2) Controls the algorithm's randomization, where different random state values (e.g., x versus y) result in varied synthetic data generation.
3) K- neighbors: Specifies the number of samples used to determine nearest neighbors for data synthesis.

Table 2. Proposed SMOTE Hyperparameter Configuration

Hyperparameter	***Values***
k_neighbors	3
random_state	70
sampling_strategy	Scenario A Class 1: 30, Class 2: 30, Class 3: 30, Class 4: 30 Scenario B Class 1: 36, Class 2: 30, Class 3: 38, Class 4: 32

An unbalanced data distribution across classes can lead to prediction bias and reduce model performance. In this study, SMOTE was applied with the proposed hyperparameter configurations, as shown in Table 2.

The k-nearest neighbors parameter is set to 3, following the rule that this value should not exceed the number of classes. The random state is set at 70. To evaluate model performance under different training scenarios, two sampling strategies with SMOTE are applied:

1) Scenario A
Each class is resampled to 30 data points, generating (30 - the original number of data points) synthetic data, totaling 84 synthetic data points.
2) Scenario B
Each class generates up to 25 synthetic data points, resulting in a total of 100 synthetic data points.

Figure *9* illustrate improved and balanced distribution of EAR feature data, enhancing the visibility of features across each class.

E. Training and Validation

The dataset is divided into three subsets: training, validation, and testing sets, each serving distinct functions.

1) Training set: Used by the model to learn patterns for making predictions.
2) Validation set: Used to evaluate model performance and adjust hyperparameters if necessary.
3) Testing set: Used to assess the model's actual performance.

The data division follows specific guidelines, with synthetic data from oversampling used for training, while validation and testing sets consist of original (non-synthetic) data and external data. Synthetic data in the training set enables the model to learn various feature combinations. The data splits for each scenario are as follows:

1) Scenario A
   According to **Error! Reference source not found.**, following the sampling strategy in section 1, there are 84 training data points, 24 validation data points, and 12 testing data points, with varied training data counts per class.
2) Scenario B
   According to , based on the sampling strategy in section 2, there are 100 training data points, 24 validation data points, and 12 testing data points, with equal training data across classes.

Training is conducted using the Multi-Layer Perceptron (MLP) algorithm with the hyperparameter configuration shown in Table *5*. An MLP is a type of artificial neural network (ANN) that consists of multiple layers of fully connected neurons. Although an MLP is a simple feedforward neural network, it is often considered part of deep learning if it includes more than one hidden layer, adding depth to its architecture [25]. MLP was chosen due to its suitability for processing non-linear data. Additionally, previous studies on drowsiness detection using EEG signal patterns have employed MLP [26], supporting its application in this study, which classifies sleepiness based on EAR patterns.

Table 3. Scenario A Class Distribution

Class	***Total***		
	Training Data	**Validation Data**	**Testing Data**
Class 1	19	8	3
Class 2	25	3	2
Class 3	17	8	5
Class 4	23	5	2
Total	84	24	12

Table 4. Scenario B Class Distribution

Class	***Total***		
	Training Data	**Validation Data**	**Testing Data**
Class 1	25	8	3
Class 2	25	3	2
Class 3	25	8	5
Class 4	25	5	2
Total	**100**	**24**	**12**

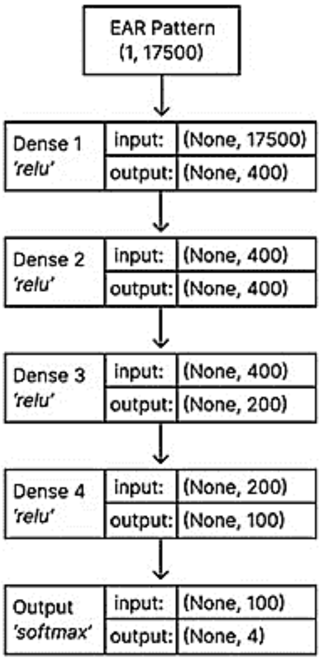


Table 5. Proposed MLP Hyperparameter Configuration

Figure 10. MLP model architecture

Hyperparameter	**Configuration**
hidden_layer_sizes	(400,400,200,100)
activation	Relu
Output activation	Softmax
Solver	Adam
Alpha	0.0001
Beta 1	0.9
Beta 2	0.999
epsilon	1e-08
Learning_rate init	0.001
Learning_rate	Invscaling
Random state	762

The model comprises six layers based on the hyperparameters in Table *5*: one input layer, four hidden layers with 400, 400, 200, and 100 neurons, and one output layer. The initial learning rate is set to 0.001 and decreases exponentially (inverse scaling). Each hidden layer uses the ReLU activation function, which outputs positive integer values, while the output layer employs the Softmax activation function to produce probability distributions and classify based on the highest probability.

The model utilizes ‘adam’ (adaptive moment estimation) as the optimizer, with alpha, beta, and epsilon configurations in accordance with the creator’s recommendations [27]. Figure *10* illustrates the architecture used for model training.

EXPERIMENTS

In this stage, the trained model is tested using a testing dataset. Testing is conducted over multiple epochs to assess the model's adequacy or determine if reconfiguration is required for improved results. Model performance is evaluated using the following metrics:

1) Confusion matrix: A matrix representing the model’s predictive performance, showing the number of correct and incorrect predictions for each class.
2) Accuracy: The ratio of correct predictions to the total number of data points, indicating overall model accuracy.
3) Precision: The number of correct predictions for a class divided by the total predictions made for that class.
4) Recall: The number of correct predictions for a class divided by the total actual data points in that class.
5) f_1 score: A harmonic mean of precision and recall, calculated as follows (Eq. 3):

$$f_1 score = 2 \times \frac{precision \times recall}{precision + recal} \quad (3)$$

6) Loss: A measurement of prediction error, using logistic/cross-entropy loss defined as follows (Eq. 4):

$$L_{log}(\ell, p) = -(\ell log(\wp) + (1-\ell)\log(1-\wp)) \ (4)$$

where ℓ is the label and $\wp$ is the estimated probability for the class.

A. Model performance analysis

Table *6* and Table *7* summarize the test results across epoch values of 60, 70, 80, 90, and 100. Scenario A uses a sampling strategy that generates a varying number of synthetic data points per class, leading to different quantities of data trained for each class. In contrast, Scenario B applies a balanced sampling strategy, resulting in an equal amount of synthetic data for each class and, consequently, the same quantity of training data per class. Based on model performance testing against the epoch value, an increase in epoch value lengthens training time. As the epochs increase, the loss value decreases while accuracy improves. A lower loss value indicates higher prediction probability, whereas a higher loss value corresponds to a lower prediction probability.

Table 6. Model Performance in Scenario A

Performance Metrics		Epoch				
		60	70	80	90	100
Training	Time	45	48	49	53	67
	Accuracy	0.78	0.97	0.976	0.98	1
	Loss	0.75	0.55	0.35	0.21	0.13
Validation	Accuracy	0.71	0.88	0.91	0.88	0.96
	Precision	0.84	0.9	0.93	0.91	0.96
	Recall	0.71	0.88	0.91	0.88	0.96
	Loss	0.89	0.79	0.52	0.39	0.27

Table 7. Model Performance in Scenario B

Performance Metrics		Epoch				
		60	70	80	90	100
Training	Time	38	42	49	49	54
	Accuracy	0.96	0.98	0.976	0.99	1
	Loss	0.65	0.44	0.32	0.17	0.09
Validation	Accuracy	0.88	0.92	0.92	0.96	0.92
	Precision	0.91	0.93	0.93	0.96	0.93
	Recall	0.88	0.92	0.92	0.96	0.92
	Loss	0.76	0.6	0.46	0.36	0.19

The MLP model, configured with four hidden layers containing 400, 200, and 100 neurons respectively, an ‘adam’ optimizer, ‘relu’ activation in hidden layers, and ‘softmax’ in the output layer, along with other hyperparameters shown in Table *5*, yields the best performance in Scenario B, where training data per class is balanced. At an epoch value of 80, this model achieves a training accuracy of 0.976 and a loss of 0.32, with validation accuracy, precision, recall, and F1 scores of 0.92, 0.93, 0.92, and 0.92, respectively, and a validation loss of 0.46. When tested on a set of 12 data points, as shown in Figure *11* the model accurately classifies each class, resulting in an accuracy of 1 and a loss of 0.46. In comparison, Scenario A, also at epoch 80, shows two misclassifications, where two data

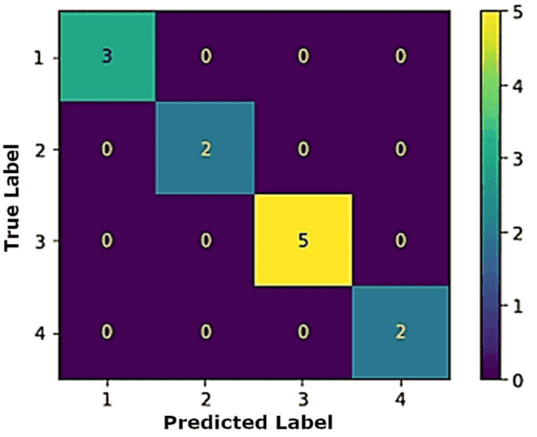


Figure 11. Model B Confusion Matrix Performance on Test Set

points from Class 3 are classified as Class 1, achieving an accuracy of 0.83 and a loss of 0.64, as shown in Figure *12*.

B. Model performance on external data

This model was also tested using external data from videos provided by volunteers. Prior to recording, subjects evaluated their sleepiness level based on the KSS standard. The model used for testing was from epoch 80, which demonstrated the best performance in both scenarios A and B.

As shown in Table 8 and Table *9*, the model classifies subject A, labeled as Class 3, as Class 4 with a prediction probability of 32.05% and a loss of 1.13 in Table 8, and as class 3 with a prediction probability of 72.65% and loss of 0.31 in Table *9*. Similarly, predictions, probabilities, and losses for Subjects B, C, and D are presented in both tables.

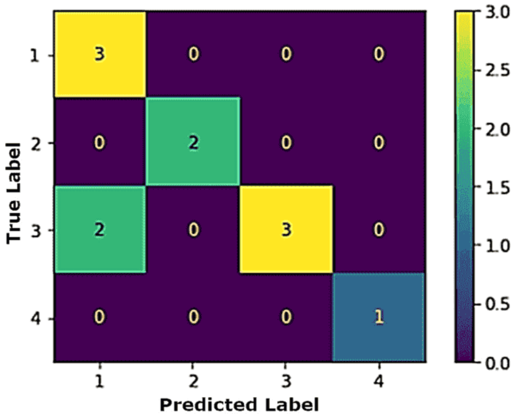


Figure 12. Model A Confusion Matrix Performance on Test Set

Table 8. Model A Performance on External Data

Video	Label	Loss	Prediction probability (%)			
			1	2	3	4
Subject A	3	1,13	25.91	10.17	31.87	32.05
Subject B	3	1.43	48.64	13.11	23.78	14.47
Subject C	3	1.13	30.16	5.7	26.26	37.88
Subject D	3	1.26	36.92	26.03	28.32	8.73

Table 9. Model B Performance on External Data

Video	Label	Loss	Prediction probability (%)			
			1	2	3	4
Subject A	3	0.31	6.83	7.52	72.65	13
Subject B	3	0.82	25.18	21.92	43.18	9.09
Subject C	3	0.52	10.3	9.62	59.31	20.78
Subject D	3	0.6	14.13	23.74	54.65	7.48

Comparing models generated in scenarios A and B, the model from scenario B exhibits lower loss and higher accuracy. This improvement is attributed to scenario B's approach of balancing synthetic data across each class, highlighting that class balance can enhance model performance.

This study also compares the proposed model with research by Han et al. [28], which examined models such as TCN, ResNet3D, and a combined TCN + ResNet3D model on datasets like UTA-RLDD, DROZY, and a custom facial image augmentation dataset. The proposed model aims to recognize fatigue through subtle changes around the operator's eyelids, using a combination of Temporal Convolutional Network (TCN) and ResNet3D. It utilizes the EAR sequence and a continuous frame eyelid image sequence to categorize the operator's mental state into three classes: sober, low-vigilance, and drowsiness.

Table 10. Comparison of the Performance of Different Trained Models

Feature	Model	Accuracy (%)
Eyelid Image sequence [28]	C3D	65.42
	Densnet3D	87.36
	Inception3D	96.38
	Resnet3D	96.53
EAR Sequence [28]	1DCNN	46.25
	LSTM128	42.92
	LSTM256	43.33
	TCN	51.11
Both eyelid Image and EAR sequence [28]	Resnet3D + 1DCNN	97.36
	Resnet3D + TCN	97.64
EAR Sequence	MLP	25
	MLP + SMOTE (*)	97.6

(*) Proposed Method

Table *10* shows that when the same MLP model was tested with the original dataset, without SMOTE oversampling, it achieved only 25% accuracy and could not classify into 4 classes. This limitation is due to insufficient data in Class 2 to be split into training, validation, and testing sets. The results indicate that the proposed framework reaches the same accuracy (97.6%) as the ResNet3D + TCN model. A notable advantage of the proposed model is that it uses fewer features, relying solely on the EAR sequence, while also accommodating a larger set of classes (4 total) compared to Han's 3 classes. In comparison with other methods that exclusively use the EAR sequence feature, the proposed model demonstrates significantly higher accuracy.

This approach uses the Eye Aspect Ratio (EAR) to detect subtle eye movements associated with drowsiness and employs a Multi-Layer Perceptron (MLP) to classify sleepiness levels. EAR serves as an effective feature for real-time eye closure detection, while the MLP's layered architecture captures complex data patterns. To address class imbalance, SMOTE generates synthetic data for underrepresented classes, enabling balanced training and reducing prediction bias. This method enhances model generalization, allowing it to classify all sleepiness levels with 97.6% accuracy, proving robust and effective for real-time drowsiness detection in practical applications.

CONCLUSION

Based on the experiments, the following conclusions can be drawn:

The Multi-Layer Perceptron (MLP) architecture with four hidden layers can classify the Karolinska Sleepiness Scale using the eye aspect ratio (EAR) value, achieving a training accuracy of 97.6% and a loss of 32%.

With the configured hyperparameters in Table *5*, the model performs optimally at an epoch value of 80. This balanced model setup, with an equal number of training data per class, yields a training accuracy of 0.976 and a loss of 0.32, a validation accuracy of 0.96 with a loss of 0.46, and a testing accuracy of 1 with a loss of 0.46.

The implementation of SMOTE to generate synthetic data effectively addressed class imbalance, significantly enhancing classification accuracy. In Scenario B, where an equal amount of training data was applied across each class, losses were lower at 0.32, compared to Scenario A, where unequal class data led to higher losses of 0.46.

REFERENCES

[1] AAA Foundation for Traffic Safety, “2022 Traffic Safety Culture Index (Technical Report),” Washington DC.

[2] J. Vicente, P. Laguna, A. Bartra, and R. Bailón, “Drowsiness detection using heart rate variability,” *Med. Biol. Eng. Comput.*, vol. 54, no. 6, pp. 927–937, Jun. 2016, doi: 10.1007/s11517-015-1448-7.

[3] Yongqiang Li, Shangfei Wang, Yongping Zhao, and Qiang Ji, “Simultaneous Facial Feature Tracking and Facial Expression Recognition,” *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2559–2573, Jul. 2013, doi: 10.1109/TIP.2013.2253477.

[4] I.-H. Choi, C.-H. Jeong, and Y.-G. Kim, “Tracking a Driver’s Face against Extreme Head Poses and Inference of Drowsiness Using a Hidden Markov Model,” *Appl. Sci.*, vol. 6, no. 5, p. 137, May 2016, doi: 10.3390/app6050137.

[5] R. Tamanani, R. Muresan, and A. Al-Dweik, “Estimation of Driver Vigilance Status Using Real-Time Facial Expression and Deep Learning,” *IEEE Sens. Lett.*, vol. 5, no. 5, pp. 1–4, May 2021, doi: 10.1109/LSENS.2021.3070419.

[6] 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. Transp. Syst.*, vol. 25, no. 7, pp. 6938–6950, Jul. 2024, doi: 10.1109/TITS.2023.3346054.

[7] G. Du, L. Zhang, K. Su, X. Wang, S. Teng, and P. X. Liu, “A Multimodal Fusion Fatigue Driving Detection Method Based on Heart Rate and PERCLOS,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21810–21820, Nov. 2022, doi: 10.1109/TITS.2022.3176973.

[8] A. Murata, T. Doi, and W. Karwowski, “Sensitivity of PERCLOS70 to Drowsiness Level: Effectiveness of PERCLOS70 to Prevent Crashes Caused by Drowsiness,” *IEEE Access*, vol. 10, pp. 70806–70814, 2022, doi: 10.1109/ACCESS.2022.3187995.

[9] P. Lancaric, N. Grilusova, and J. Kostolny, “System for Monitoring Computer User Fatigue,” in *2023 13th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Athens, Greece: IEEE, Oct. 2023, pp. 1–5. doi: 10.1109/DESSERT61349.2023.10416530.

[10] Y. Lu, C. Liu, F. Chang, H. Liu, and H. Huan, “JHPFA-Net: Joint Head Pose and Facial Action Network for Driver Yawning Detection Across Arbitrary Poses in Videos,” *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 11850–11863, Nov. 2023, doi: 10.1109/TITS.2023.3285923.

[11] Fouzia, R. Roopalakshmi, J. A. Rathod, A. S. Shetty, and K. Supriya, “Driver Drowsiness Detection System Based on Visual Features,” in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore: IEEE, Apr. 2018, pp. 1344–1347. doi: 10.1109/ICICCT.2018.8473203.

[12] 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 Syst. Appl.*, vol. 158, p. 113505, Nov. 2020, doi: 10.1016/j.eswa.2020.113505.

[13] K. Kaida *et al.*, “Validation of the Karolinska sleepiness scale against performance and EEG variables,” *Clin. Neurophysiol.*, vol. 117, no. 7, pp. 1574–1581, Jul. 2006, doi: 10.1016/j.clinph.2006.03.011.

[14] 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, Dec. 2022, doi: 10.1109/TCYB.2021.3110813.

[15] M. Dreisig, M. H. Baccour, T. Schack, and E. Kasneci, "Driver Drowsiness Classification Based on Eye Blink and Head Movement Features Using the k-NN Algorithm," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, Canberra, ACT, Australia: IEEE, Dec. 2020, pp. 889–896. doi: 10.1109/SSCI47803.2020.9308133.

[16] Y. Wu and Q. Ji, "Facial Landmark Detection: A Literature Survey," *Int. J. Comput. Vis.*, vol. 127, no. 2, pp. 115–142, Feb. 2019, doi: 10.1007/s11263-018-1097-z.

[17] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH: IEEE, Jun. 2014, pp. 1867–1874. doi: 10.1109/CVPR.2014.241.

[18] C. Álvarez Casado and M. Bordallo López, "Real-time face alignment: evaluation methods, training strategies and implementation optimization," *J. Real-Time Image Process.*, vol. 18, no. 6, pp. 2239–2267, Dec. 2021, doi: 10.1007/s11554-021-01107-w.

[19] T. Soukupova and J. ˇCech, "Real-Time Eye Blink Detection using Facial Landmarks," *21st Comput. Vis. Winter Workshop*, 2016.

[20] A. Kuwahara, K. Nishikawa, R. Hirakawa, H. Kawano, and Y. Nakatoh, "Eye fatigue estimation using blink detection based on Eye Aspect Ratio Mapping(EARM)," *Cogn. Robot.*, vol. 2, pp. 50–59, 2022, doi: 10.1016/j.cogr.2022.01.003.

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

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

[23] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

[24] Y. Bao and S. Yang, "Two Novel SMOTE Methods for Solving Imbalanced Classification Problems," *IEEE Access*, vol. 11, pp. 5816–5823, 2023, doi: 10.1109/ACCESS.2023.3236794.

[25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[26] H.-T. Nguyen, N.-D. Mai, B. G. Lee, and W.-Y. Chung, "Behind-the-Ear EEG-Based Wearable Driver Drowsiness Detection System Using Embedded Tiny Neural Networks," *IEEE Sens. J.*, vol. 23, no. 19, pp. 23875–23892, Oct. 2023, doi: 10.1109/JSEN.2023.3307766.

[27] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Jan. 29, 2017, *arXiv*: arXiv:1412.6980. Accessed: Aug. 26, 2024. [Online]. Available: http://arxiv.org/abs/1412.6980

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