Lightweight 2D-1D CNN for Real-Time Driver Drowsiness Detection with Federated Learning on Edge Devices

Guettas Amina[1], Ayad Soheyb[1], Kazar Okba[2]
[1]LINFI Laboratory, University of Biskra
[2]College of Computing and Intelligent Systems, University of Kalba, Sharjah, UAE
E-mail: amina.guettas@univ-biskra.dz, s.ayad@univ-biskra.dz, okba.kazar@ukb.ac.ae

Keywords: Drowsiness detection, fatigue detection, edge computing, federated learning, convolution neural network, eye state

Received: October 18, 2025

Driver drowsiness is one of the main leading causes of traffic crash accidents. Preventing these mishaps and limiting consequential damage and losses, whether human or material, has been the subject of many research studies these recent years. Fulfilling this task requires a reliable drowsiness detection system installed in the vehicle that can function on limited hardware such as edge devices with real-time execution speed. The present study proposes a computer vision-based driver drowsiness detection system using Convolutional Neural Networks. The traditional approach is to extract some features from the captured driver's face images such as PERCLOS, blink speed, or blink frequency, and determine the driver state using machine learning methods, predefined thresholds, or rules. However, deep learning techniques allow us to directly identify the driver state from the captured video, yet these techniques are usually computationally intensive and unsuitable for edge computing applications. To overcome these obstacles, we propose a lightweight 2D-1D CNN model for driver drowsiness detection. We deployed a 2D-CNN model for the eyes feature extraction of both the left and right eyes of the driver from every captured frame. In addition, we applied a 1D-CNN model to capture the temporal information of the concatenated features over a defined period. We implemented and evaluated the model's performance under both centralized learning and federated learning. Experimental results show that our approach achieves 99% accuracy on the DROZY dataset and 98.06% on subjects excluded from the training set in the CL, while in the FL it achieves 95.27% on the same dataset and 77.86% on unseen subjects. Furthermore, the model runs in real time on an NVIDIA Jetson Nano, reaching 25.6 FPS. While FL offers enhanced privacy by enabling local model training without sharing raw user data, it exhibits limited generalization capability. In contrast, CL provides higher accuracy and a more robust model. Comparative analysis reveals a performance trade-off for FL, balanced by improved data privacy. This work highlights the potential of privacy-preserving, real-time drowsiness detection systems using advanced deep learning architectures.

Povzetek: Študija predstavi lahek 2D–1D CNN za realnočasovno zaznavanje zaspanosti voznikov na robnih napravah, ki iz očesnih slik izlušči prostorske značilke in prek časovne konvolucije modelira dinamiko, pri čemer primerja natančnejše centralizirano učenje z zasebnostno ugodnejšim federiranim učenjem.

1 Introduction

Enhancing the safety of the road and guaranteeing the driver's safety has been the concern of both the automotive industry and governments alike. Despite the huge progress that has been made in technology, automation systems, and regulations, road accidents still cause major losses, whether human or material. According to the World Health Organization (WHO) [1], approximately 1.19 million fatalities and more than 20 million non-fatal injuries are reported around the world each year. In addition, it is estimated that about 3% of the gross domestic product (GDP) of most countries is spent on road traffic accidents. As long as humans are involved, mistakes will be made. This is especially true when performing life-or-death jobs like driving. Drowsy driving plays a major role in car accidents, contributing to 2.3% of road traffic accident fatalities in the USA and about 33,000 injuries in 2015, based on the National Highway Traffic Safety Administration (NHTSA) report [2].

A fatigue state involves a reluctance to make efforts, leading to reduced efficiency, alertness, and performance in individuals [3]. Sleepiness, fatigue, and drowsiness are widely used interchangeably in the literature. Nevertheless, sleepiness has a more specific meaning than fatigue, but they are all used to denote the sleepiness effect related to working for a certain amount of time without taking the adequate rest, which affects performance [4].

Detecting driver drowsiness can be divided into three approaches based on the nature of the input data. The first one is the physiological approach; it uses physiologi-

cal signals such as the electrocardiogram (ECG), electroencephalogram (EEG), electrooculogram (EOG), skin temperature, galvanic skin response (GSR), and electromyography (EMG). These signals are collected via electrodes placed on the driver's body. The fatigue state stimulates the different organs of the body, hence affecting the produced signals, where EEG is considered the most reliable because it records the neurophysiological activity of the human brain (electrical brain activity). However, the attached electrodes can be intrusive and uncomfortable for the driver since they can affect the driving experience negatively. The second approach takes advantage of vehicle signals. These signals are collected through sensors attached to the vehicle components such as the steering wheel and pedals, which provide information about the driving behavior, for example, steering wheel movement, lane deviation, acceleration, and braking. However, this collected information can be misleading since it is also subject to other factors such as road and weather conditions. The final approach, and the one that is used in our study, is monitoring the driver's behavior by capturing the visual features via one or multiple visual lights or near-infrared (NIR) cameras installed in the vehicle. The processing of the extracted information from the captured images such as yawning, blink speed, blink frequency, percentage of eye closure (PERCLOS), gaze region, and head pose, can help determine the driver's state. This approach is suitable for real-life applications because it is non-intrusive and does not disturb the driving experience; in addition, it gives accurate and robust results. However, it can be affected by other factors such as illumination variations, wearing glasses, and head movement [5].

Edge computing is a decentralized computing paradigm that processes data at the source instead of centralized servers. Edge AI involves deploying AI algorithms and models directly on edge devices (e.g., smartphones, IoT devices), bringing AI capabilities nearer to the data source. This reduces latency, saves bandwidth, enhances privacy, and enables offline operation. The next evolution of Edge AI is called Federated Learning (FL). FL extends beyond local inference to training models at the edge. While single edge devices lack resources for heavy model training, combining multiple devices allows decentralized training while keeping data local [6]. In the context of video-based drowsiness detection, particularly when using data from diverse user environments, FL is especially relevant. It enables on-device model training, while federated aggregation ensures that the model generalizes well without compromising individual data privacy.

This study aims to reduce driving risks and enhance road safety by proposing a real-time, vision-based driver drowsiness detection system using deep learning. We propose a complete system for drowsiness detection, starting from face localization and tracking to eye region extraction and feature extraction, ending with drowsiness classification. We focus on improving both accuracy and processing speed, as the system is intended for deployment on resource-constrained edge devices. Our approach for drowsiness detection from eye state analysis uses a hybrid 2D-1D CNN architecture that concatenates two lightweight 2D-CNN models for feature extraction and applies a 1D-CNN for drowsiness detection within a predefined time window. Additionally, we evaluate the model's performance under both centralized and federated learning settings to explore the trade-offs between accuracy and privacy. The system was evaluated and compared with previous works. Our model achieved state-of-the-art accuracy of 98.06% on unseen subjects, and 99% accuracy on the test set during the centralized training phase, alongside real-time performance of 25.6 FPS on a Jetson Nano edge device, demonstrating its robustness and efficiency under traditional centralized learning. However, accuracy decreased under the federated learning framework, achieving 95.27% on the test set and 77.86% on unseen subjects. Our findings demonstrate that federated learning offers a viable pathway for deploying real-time drowsiness detection systems while preserving user data privacy.

Our conducted experiments are guided by three core research questions. RQ1: Can a hybrid 2D–1D CNN achieve competitive drowsiness detection accuracy while remaining lightweight enough for deployment on edge devices? RQ2: How effectively does the proposed architecture generalize to unseen subjects compared to state-of-the-art approaches evaluated on the DROZY dataset? RQ3: What is the performance trade-offs between from centralized to federated learning in driver drowsiness detection? By addressing these research questions through empirical evaluation, this work contributes both a practical edge-deployable drowsiness detection system and insights into the performance-privacy trade-offs in FL.

The remainder of this paper is organized as follows. Section 2 summarizes existing and related work on driver drowsiness detection. Section 3 presents an overview of federated learning. The proposed method is described in Section 4. The experimental results and their discussion are presented in Sections 5 and 6. Finally, in Section 7, the conclusion and the potential future work are provided.

2 Related works

Many studies have been conducted on driver drowsiness detection; however, numerous challenges have been encountered, as each approach and method has its limitations. Many studies have focused on physiological signals because they are directly related to driver's physical state, making them reliable such as EEG [7], EOG [8], EMG [9], and ECG [10] with heart rate variability (HRV) and heart rate (HR) being widely used features. Furthermore, combining multiple physiological signals is also common in detecting driver drowsiness [11, 12, 13]. The main limitation of the physiological approach lies in the intrusive sensors that affect driver comfort and movement, making it difficult to deploy in real-world systems and to collect data. Furthermore, the majority of studies were conducted on small

datasets with limited subjects and mostly in laboratory-controlled environments.

Therefore, studies on non-intrusive approaches provide more realizable solutions by using vehicle-based data such as lateral deviation [14] and steering wheel [15] to determine driving behavior. However, this approach can be affected by road and weather conditions which can alter driving performance. This leads us to the last approach, processing video data to monitor driver behavior where many valuable features can be extracted from the captured images. After detecting the face, the eye or mouth regions are used to extract several features, including yawning [16], PERCLOS [17], blink frequency [18], and Eye Aspect Ratio (EAR) [19], which have been widely used by researchers. In [17], the authors calculated PERCLOS from the iris region extracted from the input video of YawDD dataset to determine driver state. First, they applied histogram equalization to adjust frame intensity to overcome changing outdoor illumination. Next, they used the Viola-Jones method to detect the face and eye regions. Then, the Hough algorithm was used for iris region extraction, and morphological filters were applied to measure iris area. Finally, PERCLOS was calculated after estimating the area. However, the accuracy was not reported. In [19], the authors obtained temporal information by applying an SVM model to classify 15 consecutive EARs into three different states (open eye, short blink, or long blink) and applied two rules to detect driver drowsiness based on model output, achieving an accuracy of 94.44% on the DROZY dataset on unseen subject. In [16], the authors employed four CNN models (FlowImageNet, VGG-FaceNet, AlexNet, and ResNet) to predict respectively behavioral features, facial expressions, environmental features, and hand gestures trained on NTHU-DDD dataset. The final state was determined using a simple averaging ensemble method, achieving an accuracy of 85% on custom dataset.

Other studies worked on detecting driver drowsiness directly from input video such as [20], where the authors proposed a 3D-CNN model to extract spatio-temporal features from consecutive video frames on the NTHU-DDD dataset, achieving 78.48% accuracy on unseen subject validation. In addition, a filter pruning sub-network was proposed to reduce the inference time. The authors in [21] applied two CNN models to extract fatigue feature representations from the eyes and mouth and fed the fused features using a factorized bilinear feature fusion model into a long short-term memory (LSTM) to capture temporal information and determine driver state. Their approach achieved 75.8% accuracy on NTHU-DDD on unseen subject. In [22], the authors developed an ADAS (advanced driver assistance system) for driver drowsiness detection using a sequence of 600 frames. Two systems were proposed and compared: one based on LSTM and the other on fuzzy logic, using YawDD and UTA-RLDD datasets. The first system applied a recurrent CNN (combination of RNN and CNN) for driver state classification. The second system combined AI techniques and a CNN model for blink and yawning detection with a fuzzy inference system. The reported accuracy was approximately 60% on the UTA-RLDD test data across 5-fold cross-validation.

Nevertheless, these studies did not take into consideration the processing time and the resources required for optimal system functioning. These parameters are important to consider in driver drowsiness detection, where systems must be lightweight, installed in-vehicle, and capable of making real-time decisions.

Therefore, other studies on driver drowsiness detection systems oriented toward edge computing and mobile devices have been conducted. The particularity of edge devices is their limited computational and memory resources, which make it challenging to develop real-time systems that do not require heavy computational power and achieve high accuracy. In [23], the authors proposed an optimized model for the edge computing device NVIDIA Jetson Nano that can run up to 58 FPS. The system first locates the face and classifies it as "normal" or "distracted" based on CNN and face alignment. Moreover, features such as EAR, MAR, and head pitch angle are extracted from the "normal" state to detect driver fatigue. The other state "distracted" is used to detect driver distraction. The accuracy achieved by their system is 89.55% across multiple datasets, including YawDD, AFLW and DriverEyes, although it was not explicitly reported if it was tested on unseen subject. In [24], a CNN-based system was developed and trained on YawDD and custom datasets. The system was tested on a Raspberry Pi 4 device, achieving 94.7% accuracy on real driving environment and a speed of 10.4 FPS which is considered a real-time speed for an embedded system according to the authors. It consisted of two CNN stages, one for the facial feature extraction and eye and mouth region localization. The other network has the role of classifying the state of the eyes and mouth into "open" and "closed". In [25], the authors proposed a lightweight CNN for driver drowsiness detection system from facial, eye, and mouth regions and implemented it on a smartphone device, achieving 94.39% accuracy on custom dataset on unseens subject and real-time performance with 60 FPS.

As FL emerged as a new paradigm, a number of studies addressed it in the context of driver drowsiness detection. In [26], the authors proposed a FL framework for drowsiness detection. Employing 3D-CNN and 2D-CNN with the FedAvg strategy, they achieved 90.1% and 99.2% accuracy, respectively on the YawDD dataset. In [6], the authors compared the federated and centralized approaches with the same CNN parameters for drowsiness detection on DDD dataset using non-Independently and Identically Distributed (non-IID) data. The centralized model reached 83% accuracy, while the federated model achieved 66%. However, these studies on FL did not evaluate unseen subjects nor consider edge deployment.

3 Overview of federated learning

Federated Learning (FL) is a distributed machine learning paradigm that was introduced by Google [27, 28, 29]. Its main idea is to enable multiple clients or edge devices to collaboratively train a common model without sharing their local data. Unlike traditional centralized learning methods, where data is aggregated on a central server for training, FL ensures that raw data stays on local devices by only exchanging the model parameters. This approach significantly enhances data privacy and security, particularly in applications involving sensitive or confidential information. It also reduces communication latency and efficiently manages large amounts of data [30].

The standard FL training process consists of multiple rounds of communication between client devices and a central server. During each round, the global model is distributed to the clients, who then train the model locally using their private data. Next, each client sends model updates to the central server, which aggregates them to create an updated global model via an aggregation algorithm such as federated averaging. The global model is evaluated and redistributed to the clients for the next training round. This process repeats iteratively across multiple communication rounds to improve the global model [31]. The Figure 1 shows the process of FL training.

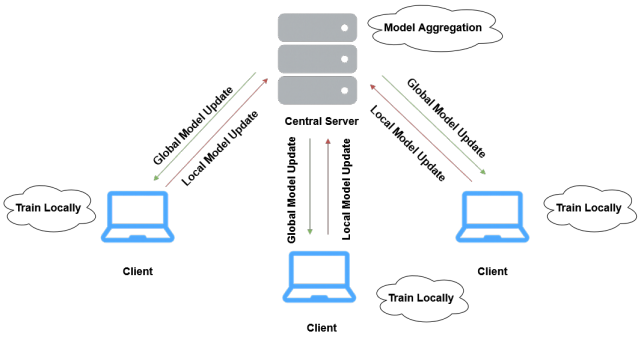

Figure 1: Overview of the federated learning training process

FL has several advantages compared to traditional centralized learning, namely privacy preservation, as only model updates are shared with the server while raw data remains on client devices. Additionally, the transfer of large datasets is greatly reduced in FL, thus lowering bandwidth requirements and improving data efficiency. Enhanced security is another advantage; by keeping data on client devices, the risk of data leakage and unauthorized access is reduced. Moreover, FL can enable faster inference and lower latency on edge devices in some scenarios, improving overall performance [6].

Despite its benefits, FL introduces unique challenges, including expensive communication due to limited bandwidth, energy, and power. To address this issue, communication-efficient methods that minimize both the number of communication rounds and the size of sent messages need to be developed. Another challenge is the heterogeneity of the system, since devices can vary in hardware, network connectivity, and power. Additionally, only a limited number of devices is active at any time, requiring FL methods to handle low participation, heterogeneous hardware, and device failures. Furthermore, statistical heterogeneity can also be considered a challenge where data across devices is non-identically distributed, with varying data size. Moreover, although FL avoids sharing raw data, transmitting model updates may still pose privacy concerns regarding the leakage of sensitive information. Techniques like secure multiparty computation (SMC) and differential privacy improve privacy but may reduce model performance or efficiency, requiring careful trade-offs [32].

The aggregation strategy plays a critical role in FL. The most used method is Federated Averaging (FedAvg), which combines the model updates from all clients using averaging. While FedAvg is effective in many scenarios, it may struggle with non-IID data, prompting the development of alternative strategies such as Weighted Average, FedProx, FedAdapt, and FedPer, which aim to provide better adaptability and customisation for the unique requirements of non-IID datasets and learning scenarios [6].

4 Proposed method

Our proposed system detects the driver's face region Fr_i from each frame $f_i \in [f_1, f_2, \ldots, f_T]$ where T denotes the number of frames in the video, using a Haar-Cascade algorithm at first, and then a correlation tracking algorithm is applied to keep track of the face region when the head position changes. Additionally, using the Dlib [33] pre-trained facial landmarks detector, the system determines 68 facial coordinates, as shown in Figure 3, in order to extract the eye regions. After the left L_i and the right R_i eye images, each of size 52 x 52 pixels, are extracted and normalized by dividing by 255, they are fed into our 2D-CNN model, and then the resulting feature vectors X_i and Y_i are concatenated into a vector V_i, which is added to a set Seq containing a certain number N of concatenated feature vectors from previous frames. This sequence is the input of the 1D-CNN model, which is used to capture temporal changes and determine the driver's state $S_k \in \{S_1, S_2\}$. The full architecture of the proposed system is illustrated in Figure 2, and Algorithm 1 describes the overall system.

In this section, we describe the architecture of the proposed system and explain its functionality. Moreover, we define the structure of the CNN models used for eye feature extraction and for determining the final driver state.

4.1 Face detection and tracking

For face region detection, we used the well-known Haar-cascade classifier for its efficiency and fast processing time. While modern deep learning face detection models like RetinaFace [34] offer higher accuracy and greater robustness than the traditional Haar cascade, they also come with

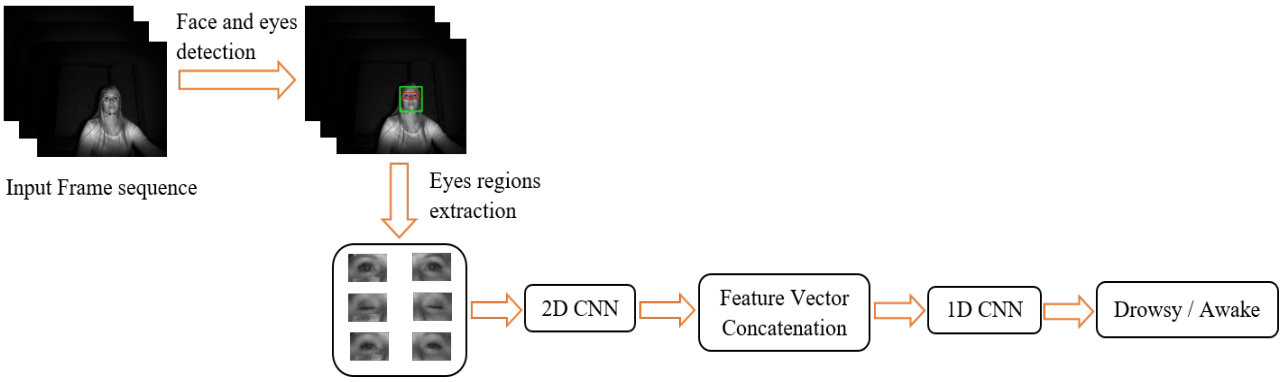

Figure 2: The architecture of the proposed system

Algorithm 1 Driver Drowsiness Detection System

Input: $Video = [f_1, f_2, ..., f_T]$
Output: $S_k \in \{S_1, S_2\}$
  while $i \leq T$ **do**
    if Fr was not detected previously **then**
      $Fr_i \leftarrow faceDetection(f_i)$
    else
      $Fr_i \leftarrow faceTracking(f_i, Fr)$
    end if
    $L_i, R_i \leftarrow eyeAreaExtraction(Fr_i)$
    Reshape L_i and R_i to $52 \times 52 \times 1$
    Normalize L_i and R_i by dividing by 255
    $X_i, Y_i \leftarrow 2D\text{-}CNN(L_i, R_i)$
    $V_i \leftarrow Conctenate(X_i, Y_i)$
    Add V_i to Seq
    if the length of $Seq = N$ **then**
      $S_k \leftarrow 1D\text{-}CNN(Seq)$
      Empty(Seq)
    end if
  end while

higher computational cost [35]. In contrast, the Haar cascade remains well suited for lightweight, real-time detection on embedded hardware, making it a practical choice when efficiency and resource usage are priorities. This algorithm is based on the Viola-Jones algorithm [36] and is implemented in the OpenCV library. The classifier is made using Haar-like features, which are used to build weak classifiers capable of detecting the edges and lines of the face. Furthermore, the calculation of the integral image (see Equation 1) reduces the overall complexity and the processing time. These weak classifiers are combined using the AdaBoost algorithm to form a strong classifier capable of detecting face regions.

$$ii(x, y) = \sum_{(x' \leq x, y' \leq y)} i(x', y') \tag{1}$$

Where $ii(x, y)$ is the integral image at location x, y, it computes the sum of the pixel values above and to the left of x, y, including $i(x, y)$.

After the face region is detected for the first time, a face tracking algorithm is applied to follow head movements and prevent losing the face trace between frames or under facial occlusions, since using the Haar-Cascade classifier alone did not yield good results in this context. Hence, we deployed the Dlib implementation of a correlation tracking algorithm [37] for this task. The tracker learns discriminative correlation filters, based on a scale pyramid representation, for separately estimating translation and scale, which provides improved performance compared to certain tracking-by-detection approaches [38]. Additionally, its computational efficiency makes it well-suited for deployment on edge-oriented hardware.

4.2 Eye extraction

After face localization, the left and right eye patches are extracted via identification of the positions of 68 (x,y) facial coordinates. Their position is illustrated in Figure 3 [39]. Furthermore, these coordinates are detected using the Dlib pre-trained facial landmarks detector that is based on an ensemble of regression trees [40]. The facial landmarks position allow the localization of a few important facial regions such as the eyes, nose, and mouth. However, we are only interested in the position of the facial landmarks of the left and right eye regions in this study.

The eye patches are scaled to 52×52 pixels to reduce the computational power needed by the CNN model to process each image and thereby reduce the execution time of the system. In addition, it maintains the useful information that can be extracted from the eye images.

4.3 Convolutional neural network

Convolutional Neural Network (CNN), also called ConvNet, is a Deep Neural Network (DNN) algorithm that is widely used for 2D data classification, especially for tasks such as image recognition , due to its strong performance across various domains. DNNs are neural networks with multiple hidden layers capable of feature extraction directly from the input by selecting the most relevant features, reducing the need for human expertise, unlike traditional al-

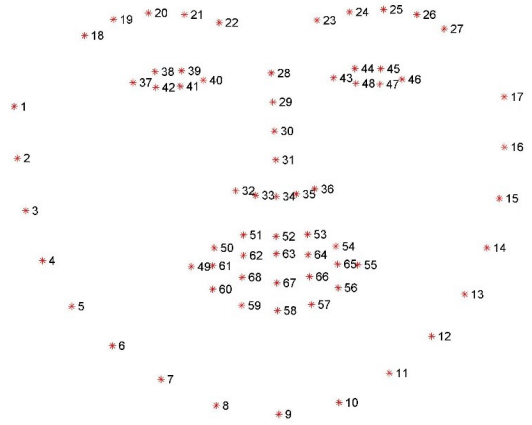

Figure 3: 68 facial landmarks position

gorithms that require a dedicated feature extraction phase. The CNN's structure consists of convolutional, pooling, and fully connected layers. The convolutional layer is its main component; it computes the feature map from the previous layer's output using multiple trained convolution filters to obtain increasingly high-level feature representations, as expressed in Equation 2 [41].

$$I_i^l = (\sum_j I_j^{l-1} \otimes w_{ij}^l + b_i^l) \tag{2}$$

Where I_i^l is the output matrix, $\otimes$ is the convolution operator, w_{ij}^l and b_i^l represent the convolutional kernels and the bias value respectively.

An activation function is applied to the result; in our case, we used Rectified Linear Unit (ReLU) activation function to address the vanishing gradient problem. It simply discards negative values and preserves positive ones, as shown in Equation 3.

$$f(x) = max(0, x) \tag{3}$$

The pooling (or subsampling) layer typically follows the convolutional layer and reduces the dimensionality of the feature maps, thereby reducing the number of trainable parameters and the computational cost of the network. The most commonly used pooling operations are max pooling, which returns the maximum value in the pooling region, and average pooling, which returns the average value in the pooling region. The max pooling formula can be expressed as follows [42]:

$$y_{ij} = max_{(pq)\in P_{ij}} x_{pq} \tag{4}$$

Where P_{ij} is the pooling region around (i, j), y_{ij} is the value at (i, j) of the output feature map and x_{pq} is the value of the input map at (p, q).

Finally, the feature maps are reshaped into a one-dimensional vector using an operation called flattening and fed to the fully connected (FC) layer for feature combination and classification. The output of each element i' in the fully connected layer is described using the following Equation 5 [43]:

$$y_{i'} = \sum_i W_{ii'} x_i + b_{i'} \tag{5}$$

Where x_i denotes feature i of the input vector, W is the weight matrix, and b represents the bias.

The last fully connected layer is the output layer, it calculates the probability distribution of each class by applying the Softmax function $s : \mathbb{R}^k \rightarrow \mathbb{R}^k$ which is defined in Equation 6 below:

$$s(z)_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}}, \quad for\ i = 1, ..., k\ and\ z = (z_1, ..., z_k) \in \mathbb{R}^k \tag{6}$$

Where z_i is an element of the input vector z and k is the number of classes.

2D-CNN and 1D-CNN are both widely used CNNs. The main difference between the two is the shape of the input and the size of the kernels. 2D-CNN accepts two-dimensional data in the shape of matrices such as images and 1D-CNN inputs are one-dimensional sequential data such as signals. The kernels or the convolutional filters are also two-dimensional matrices for the 2D-CNN and one-dimensional vectors for the 1D-CNN. Moreover, 2D-CNN is effective in feature extraction, especially image recognition, and 1D-CNN is mainly used for time series classification. In addition, 1D-CNN requires less computational power in comparison with 2D-CNN, where the computational complexity of 2D convolution can be represented as $O(N^2K^2)$ for $N \times N$ image and kernel of size $K \times K$ while the same 1D convolution is described as $O(NK)$ for input and kernel sizes of N and K, respectively [44]. The significant difference in the computational complexity makes 1D-CNN suitable for mobile and edge device hardware real-time applications.

Therefore, to create a real-time driver detection system for edge devices and to capture both spatial and temporal information, we proposed two lightweight CNNs and trained them separately. The first one is 2D-CNN for eye state classification; its architecture is illustrated in Figure 4 and Table 1 describes the output and number of parameters for each layer. Its input is a grayscale image of the eye region of size 52 x 52 x 1, followed by two convolutional blocks consisting of a convolutional layer with small (3 x 3) kernels to reduce the number of parameters, and thus the computational complexity, with a ReLU activation function and a max pooling layer with filters of size (2 x 2). For the final classification, a fully connected layer of size 32 is applied and a Softmax classifier to give the output over 2 classes (open and closed).

The second proposed network is a 1D-CNN for drowsiness detection from a sequence of frames as described in Figure 5 and Table 2. The input is a concatenated feature vector of size 64 extracted from a predefined number of frames. The model consists of a 1D-convolutional layer with small kernels of size 3, followed by a max pooling

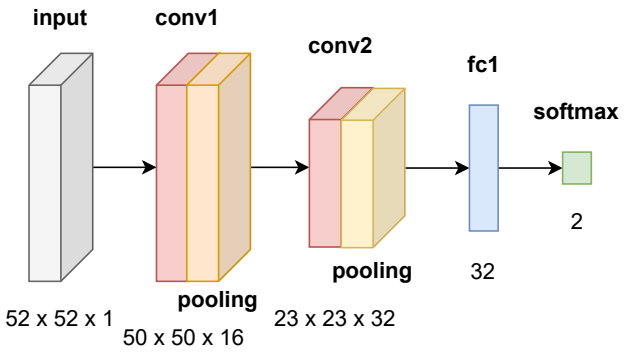

Figure 4: Structure of the eye state detection 2D-CNN model

Table 1: 2D-CNN model architecture summary

Layer	Output Shape	Param
Conv2D	50 x 50 x 16	160
MaxPooling2D	25 x 25 x 16	0
Conv2D	23 x 23 x 32	4640
MaxPooling2D	11 x 11 x 32	0
Flatten	3872	0
Dense	32	123936
Softmax	2	66
Total params: 128,802 (503.13 KB)		

layer with filters of size 2, followed by global max pooling, a FC layer of size 2, and a Softmax activation function to classify the driver's state into drowsy or awake. We chose 1D-CNN for time series classification because it is a lightweight model compared to RNN and LSTM and it has proven to achieve high accuracy with fast execution time.

Both of these CNNs were used to create a hybrid 2D-1D CNN for driver drowsiness detection. The pipeline of the 2D-1D CNN and the training process of each model are described in the following part.

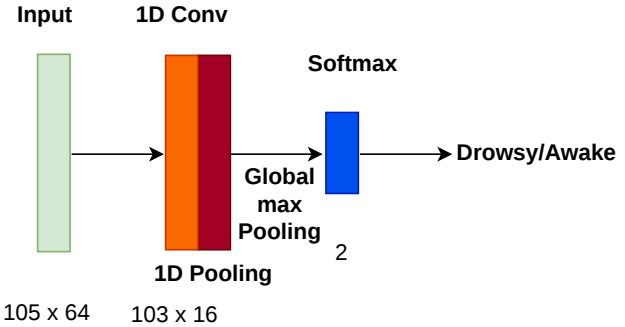

Figure 5: The architecture of driver drowsiness detection 1D-CNN model

Table 2: 1D-CNN Model architecture summary

Layer	Output Shape	Param
Conv1D	103 x 16	3088
MaxPooling1D	51 x 16	0
GlobalMaxPooling1D	16	0
Softmax	2	34
Total params: 3122 (12.20 KB)		

4.4 Eye feature extraction

The eye state is one of the best indicators of the driver's drowsiness. It is expected for the driver to keep their eyes open throughout the driving time except for the blinking interruption where increased blink frequency and eye closure duration indicate a higher likelihood that the driver is drowsy [45]. Hence, the blink frequency, duration, PERCLOS, and other features related to the eyes are usually calculated to identify the sleepiness of the driver. Therefore, to capture the change in the driver's eye states, we proposed a 2D-CNN model shown in Figure 4, and trained it to classify the eye state into open and closed. Nevertheless, we are working on the driver's drowsiness detection from the video where it is vital to capture the temporal aspect and not only the spatial one. Thus, the state of each eye separately at any given moment does not contain sufficient information to determine whether the driver is drowsy or awake. Therefore, we are interested not in the final eye state but in its feature vector so we dropped the fully connected layers from the trained 2D-CNN model of eye state detection since their main purpose is the classification. This feature vector is extracted by adding a global average pooling (GAP) layer after the last convolutional layer of our model.

GAP was proposed in [46] in order to replace the fully connected layers, where the traditional approach is to transform the feature maps generated from the last convolutional layer into a one-dimensional vector and feed it to the fully connected layers which lack direct spatial correspondence between features and output classes. However, GAP regularizes the structure of the network by creating a correspondence between each class and each feature map, resulting in the same number of classes and feature maps. The advantage that global average pooling brings over a normal flatten layer and FC layers is the dimensionality reduction by calculating the average of each input feature map, which is essential for a lightweight model to run on edge devices. In addition, it does not require any parameter optimization, unlike the FC layer, which prevents overfitting.

By replacing flatten with a GAP layer, the output feature vector was reduced from size 3872 to size 32, improving its performance and significantly reducing the number of parameters and the inference time. The functioning principle of the GAP layer in our model is illustrated in Figure 6.

In addition, we concatenated the results of two of the same 2D-CNN models, one for the left and one for the right eye feature extraction. The feature concatenation process is defined in Equation 7. Given two feature vectors

$x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$, where $x = [x_1, x_2, \ldots, x_n]$ and $y = [y_1, y_2, \ldots, y_m]$. The concatenation of the vectors x and y is a vector:

$$h = [x, y], where\ h \in R^{n+m} \tag{7}$$

In our case, each model takes the right and left eye patches as input and outputs a feature vector of size 32. Their results are concatenated into one feature vector of size 64. The concatenated model allows extraction of the spatial features vector of both eyes at the same time for each captured frame. The rationale behind it is to let the model learn to classify the driver state from both eyes simultaneously at each frame, which is less prone to misclassification than using each eye separately. In addition, it provides better interpretation of the driver's state at a specific moment.

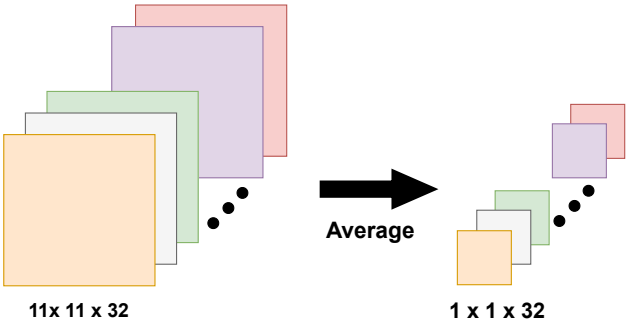

Figure 6: Global Average Pooling schema

4.5 Driver drowsiness detection

The temporal analysis of video data requires more than the spatial information of one frame. Therefore, we added the concatenated vector to the other concatenated vectors of each 15 frames from every second over a certain period, resulting in a sequence of N concatenated feature vectors of shape (N, 64), details are provided in the DROZY database section. The number of these concatenated vectors depends on the duration chosen for the experiment. We fed them to our 1D-CNN model that is illustrated in Figure 5 for driver drowsiness state classification. Figure 7 illustrates the full architecture of the proposed 2D-1D CNN.

5 Experimental results

In this section, we present our experimental results. The experiments were conducted using the publicly available DROZY and MRL datasets. The training hyperparameters, and federated learning configuration are described below. The proposed system was implemented using standard libraries (OpenCV, Dlib, TensorFlow, Keras), and the federated learning setup was trained using the Flower framework.

5.1 MRL eye dataset

To train the 2D-CNN model for eye state classification, we used MRL eye dataset [47]. It is a large-scale dataset containing 84,898 infrared eye images from 37 subjects (33 men and 4 women) captured under different lighting conditions. Additionally, it provides various information about each image such as gender, glasses, eye state, and lighting conditions.

We divided the dataset into three subsets of 70%, 15%, and 15% for training, validation, and testing, respectively. The model was compiled using the Adam optimizer [48] with a learning rate of 0.001 and a categorical cross-entropy loss function defined in Equation 8 [49] with a batch size of 64. We achieved an accuracy of 99% on both the validation and test sets using the proposed 2D-CNN model described previously.

$$CE = -\sum_{i=1}^{m} \tilde{y}_i log(y_i) \tag{8}$$

where m is the number of classes, y_i is the $i-th$ prediction value of the network, and $(\tilde{y}_i)$ is the $i-th$ ground truth label of the training samples.

5.2 DROZY database

In order to train and test our model on a video dataset, we chose the DROZY database [50] (The ULg Multimodality Drowsiness Database). It includes videos, physiological signals (EEG, EOG, ECG, and EMG), and other drowsiness-related data, recorded from 14 subjects (3 males, 11 females). These subjects were asked to perform three psychomotor vigilance tests (PVTs) on two consecutive days. Each test lasted 10 minutes and occurred under cumulative sleep deprivation conditions.

The PVT is a widely used test that measures the reaction time (RT) of participants to press a button when a visual stimulus randomly appears, and if the RT exceeds a certain value, it is considered a lapse. In addition to PVTs, a subjective metric, Karolinska Sleepiness Scale (KSS) [51] score was filled out by the participants before each PVT. The KSS is a well-known self-evaluated indicator of drowsiness level and has been shown coherent with drowsiness and is commonly used by researchers. It consists of nine levels of sleepiness, ranging from 1 (extremely alert) to 9 (very sleepy, great effort to keep awake, and fighting sleep). Table 3 [52] demonstrates the nine levels of KSS with their associated labels.

The DROZY database videos were recorded using a single near-infrared (NIR) camera with a resolution of 512 x 424. However, the database did not include all the videos of the three PVTs for all 14 subjects, which reduces the useful data that can be exploited in our experiment. In addition, two different frames per second (FPS) were used to record the videos (15 and 30 FPS); to overcome this problem, we adopted 15 FPS as the standard frame rate for processing all videos by converting those of 30 FPS to 15 FPS by skipping every other frame. Moreover, following [19], we labeled

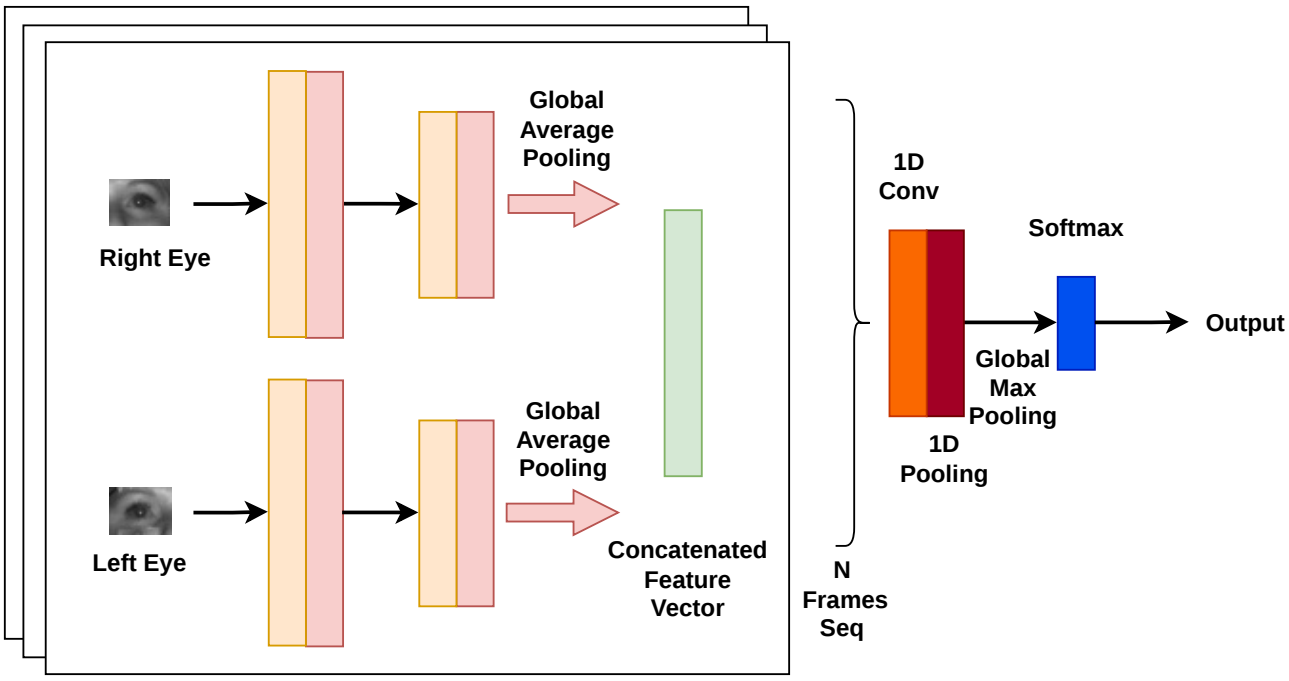

Figure 7: The architecture of driver drowsiness detection 2D-1D CNN model

the first PVT as ”Alert” and the third PVT as ”Drowsy,” excluding the second PVT. This choice was made to remove the nuance between the two states and to avoid ambiguity in labeling intermediate drowsiness states since the duration of sleep deprivation needed to reach the drowsy state may differ among individuals. Thus, the second PVT cannot be surely considered as drowsy in contrast to the third PVT. Furthermore, to ensure accurate labeling of the videos, we took into consideration the self-evaluation KSS score of each subject. We only used videos with KSS < 4 for the awake state and KSS > 6 for the drowsy state because the scores between these intervals might not be evident regarding what should be labeled and since this is a subjective evaluation based on the self-estimation of each subject, the scores may not accurately reflect the drowsiness level. Table 4 shows the subjects and their corresponding KSS scores (those in bold are the ones used in this study and those without scores are not provided in the dataset). Subjects 1, 5, 6, 8, and 10 have PVT1 and PVT3 data that met our constraint, so we used them to train and test the proposed model whereas the remaining videos of other subjects were used to evaluate its performance on unseen subject's videos, except for subject 13. The videos of subject 13 were excluded from both training and testing because his KSS scores did not fall within the defined interval.

5.3 Transfer learning

In general, training deep learning models requires a large amount of data. However, in many domains, finding and collecting such a quantity of data is not easy and manageable due to time, cost, or availability for example, medical data. To overcome this problem, researchers resort to fine-tuning other pre-trained models from related domains and adapting them to perform the desired task. Many well-known models are publicly available for use such as VGG 16 [53] and AlexNet [54]. Yet, these models can be resource-intensive and unsuitable for mobile and embedded systems, therefore we propose our lightweight 2D-CNN model for eye state detection, shown in Figure 4. We randomly extracted 100 eye images from videos of subjects 1, 5, 6, 8, and 10 that were used to train the drowsiness model and labeled them as open or closed. However, the number of extracted frames was insufficient for training a deep learning model so we resorted to using the MRL eye dataset for training and the extracted frames of the DROZY dataset for fine-tuning the model to improve eye state detection accuracy, given the differences between the two datasets, such as image quality and camera angle.

Like the MRL dataset, the collected frames from the DROZY dataset were divided into 70% for training, with the remaining 30% split equally between validation and test sets. We explored different fine-tuning configurations and concluded that full model training was the best configuration for it to converge. Freezing all layer weights except the top layers, or partially freezing the first few layers, did not perform well. The model achieved an accuracy of 95% on the test set.

5.4 Centralized learning results

Videos from subjects with KSS scores of PVT1 < 4 and PVT3 > 6 were used to train our 2D-1D CNN model for drivers' drowsiness detection. Only 5 subjects met these

Table 3: The Karolinska sleepiness scale (KSS)

Scale	Description
1	Extremely alert
2	Very alert
3	Alert
4	Rather alert
5	Neither alert nor sleepy
6	Some signs of sleepiness
7	Sleepy, but no effort to keep awake
8	Sleepy, some effort to keep awake
9	Very sleepy, great effort to keep awake, fighting sleep

Table 4: The subjects and their corresponding KSS scores (The PVTs in bold are the ones used in training and testing)

Subject	KSS PVT 1	KSS PVT 3
1	**3**	**7**
2	**3**	6
3	**2**	4
4	4	**9**
5	**3**	**8**
6	**2**	**7**
7	-	**9**
8	**2**	**8**
9	-	**8**
10	**3**	**7**
11	4	**7**
12	**2**	-
13	6	-
14	5	**8**

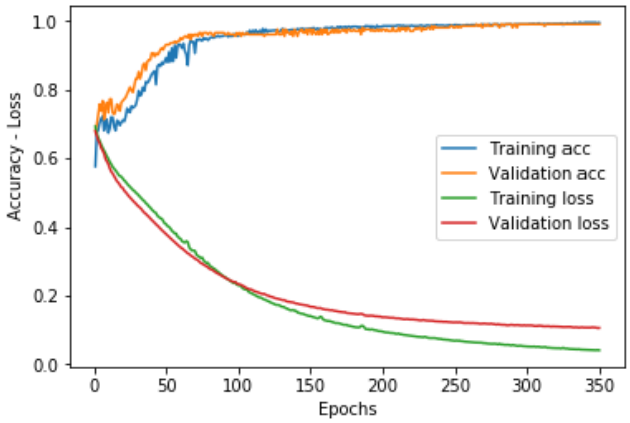

Figure 8: The 2D-1D CNN model's accuracy and loss graph during the training process

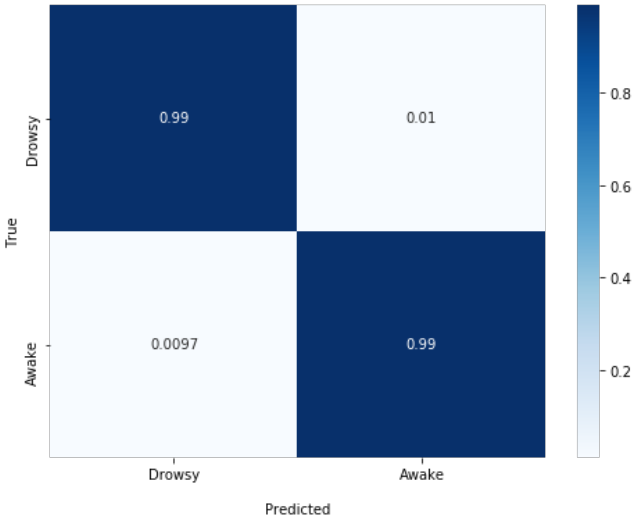

Figure 9: The confusion matrix of the 2D-1D CNN model in the test subset

constraints, which were subjects 1, 5, 6, 8, and 10 so we divided the 10 minute videos from each PVT into sequences of a defined duration. These sequences were labeled as awake or drowsy according to the PVT number and randomly divided into three subsets for training, validation, and testing at proportions of 70%, 15%, and 15%, respectively. The model was compiled using Adam optimizer with a learning rate of 0.001 and a categorical cross-entropy loss function with a batch size of 64. We achieved 99% accuracy on both the test and validation sets using 7-second sequences, demonstrating the model's efficiency. The changes in accuracy and loss during model training are reported in Figure 8.

The confusion matrix for the two classes "Drowsy" and "Awake" on the test set is presented in Figure 9. The results show 99% accuracy for both classes, suggesting that the proposed model rarely misclassifies the input.

Test on unseen subjects

The remaining subjects, which were not seen during training, were used to evaluate the model's robustness in the second phase. By keeping the constraints of the KSS score of PVT1 < 4 and the KSS score of PVT3 > 6, we were left with videos of subjects 2, 3, 4, 7, 9, 11, 12, and 14. These videos were also partitioned into small sequences of a defined duration in order to test the model. Table 5 shows the accuracy results for each subject along with their corre-

sponding KSS scores when using 7-second sequences.

Table 5: The accuracy of drowsiness detection on the test subjects' PVTs and their corresponding KSS scores

Subject	PVT	KSS	Accuracy
2	1	3	100%
3	1	2	100%
4	3	9	96.42%
7	3	9	97.61%
9	3	8	90.58%
11	3	7	100%
12	1	2	100%
14	3	8	100%

The test results demonstrate the efficiency of our model with five of the eight subjects achieved an accuracy of 100%, and the remaining subjects achieved accuracies over 90%. It is important to note that no video segments from these subjects were used during training, which demonstrates that the model performs well on unseen data and shows strong generalization.

Furthermore, to identify the best input sequence length in driver drowsiness detection and to study the impact of sequence length on the results, we divided the videos into sequences of varying lengths based on the number of seconds. The results are presented in Figure 10, showing the overall accuracy of the proposed 2D-1D CNN model on unseen subjects' videos according to sequence length in seconds.

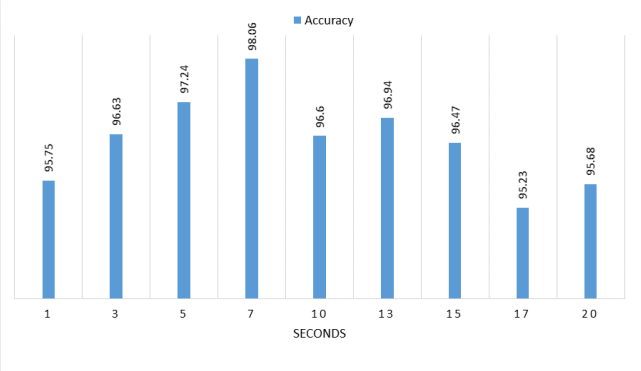

Figure 10: The accuracy of the 2D-1D CNN model according to input sequence lengths

Contrary to our intuition, increasing sequence length does not improve model accuracy. We expected that longer sequences would contain more information about driver state and that training on these sequences would make the model more accurate and robust. However, according to the results shown in Figure 10, the accuracy gradually improved, reaching its peak at 7-second sequences, and then decreased thereafter. Therefore, we adopted 7-second sequences in our system, as they proved to be the best time window.

Moreover, to demonstrate the effectiveness of our method, we ran additional tests. We used the same pipeline without concatenating the two 2D-CNN models and fed the feature vectors of size 32 from each 2D-CNN directly to the 1D-CNN and trained it using the same method with an input size of (210,32). The results on unseen subjects' videos showed a higher accuracy achieved by our model using 7-second frame sequences. Furthermore, the accuracy also regressed when the hybrid 2D-1D CNN is applied without fine-tuning the eye state detection 2D-CNN model on the DROZY dataset. Furthermore, we evaluated the model performance using only one eye by duplicating it to serve as both the left and right eye inputs. These results are shown in Table 6.

In addition, we compared the obtained results with previous works conducted on the DROZY dataset that classified two classes and were evaluated on unseen subjects. Despite the different data processing approaches in each study, our model achieved state-of-the-art accuracy of 98.06% on unseen subjects, alongside a 99% accuracy on the test set. Table 7 summarizes the previous studies and their attained results.

5.5 Federated learning results

The same videos of the 5 subjects with KSS scores of PVT1 < 4 and PVT3 > 6 were used in our FL framework to train our 2D-1D CNN model for driver drowsiness detection. In this setup, each subject was treated as an individual client in the federated learning (FL) framework. The 10-minute videos of each PVT were segmented into sequences of a defined number of seconds and labeled as 'awake' or 'drowsy' based on the corresponding PVT number. The same 2D-1D CNN architecture with identical training parameters (Adam optimizer with a learning rate of 0.001, categorical cross-entropy loss function, and batch size of 64) was used for local training on each client. Federated Averaging (FedAvg) was employed as the aggregation algorithm, with a total of 300 communication rounds and 5 local training epochs per client in each round. The aggregated global model achieved an accuracy of 95.27% on the test set using 7-second sequences, demonstrating the effectiveness of our federated approach. The accuracy and loss throughout the training process are shown in Figure 11.

The FL training curves in Figure 11 demonstrate stable convergence. Validation accuracy improves steadily while validation loss decreases consistently across training rounds. This indicates effective federated aggregation despite data heterogeneity, where each client represents a single subject. Table 8 presents the model accuracy on each client's local data.

The global model achieved high accuracy on most clients, with four clients achieving above 94%. Client 4 recorded the lowest accuracy at 85.29%, which may reflect data heterogeneity across subjects, a common challenge in FL settings where each client's local data distribution differs. Overall, the global model demonstrates reasonable gener-

Table 6: The accuracy of the tested models on unseen subjects

Model	Awake state	Drowsy state	Total
2D-1D CNN (Our model)	100%	96.91%	98.06%
2D-1D CNN without concatenation	100%	91.21%	94.5%
2D-1D CNN without fine-tuning	96.82%	94.29%	95.24%
2D-1D CNN using one eye	92.16%	90.47%	91.53%

Table 7: Comparison of our model with previous works on the DROZY dataset

Model	Accuracy	Classes	Sequence length
2D-1D CNN (Our model)	99% (test set) 98.06% (unseen subjects)	2	7s
SVM + rules [19]	94.44% (unseen subjects)	2	0.7s input, 60s window
MobileNet + logistic function [55]	91.13% (test set) 86.21% (unseen subjects)	2	Not reported (10s on Android prototype)

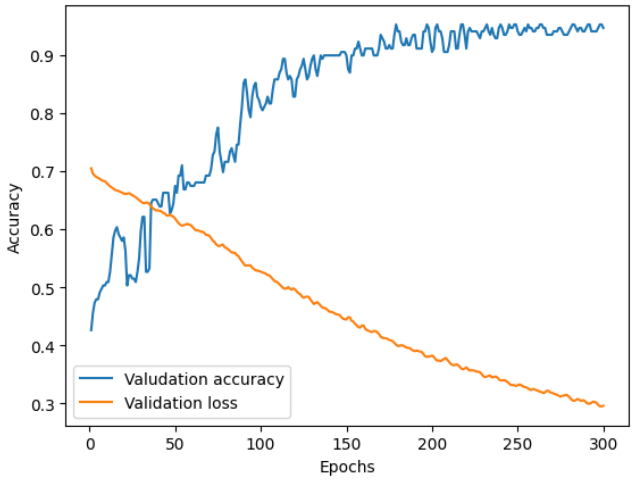

Figure 11: The 2D-1D CNN model's validation accuracy and loss graph during the FL training process

Table 8: The accuracy of the global model on the clients' local data

Client ID	Trained on subject	Accuracy
1	1	94.11%
2	5	100%
3	6	94.11%
4	8	85,29%
5	10	100%

alization across clients.

Test on unseen subjects

The global FL model was tested on the unseen subjects videos during training to evaluate its robustness in the second phase. Similar to centralized training, we used videos of subjects 2, 3, 4, 7, 9, 11, 12, and 14 that met the constraints of KSS scores of PVT1 < 4 and PVT3 > 6. These videos were also partitioned into small sequences of 7-second sequences in order to test the model. The model achieved a total accuracy of 77.86%. The confusion matrix for the two classes "Drowsy" and "Awake" is presented in Figure 12. The results show a balanced classification exceeding 74% accuracy for both classes. Table 9 shows the

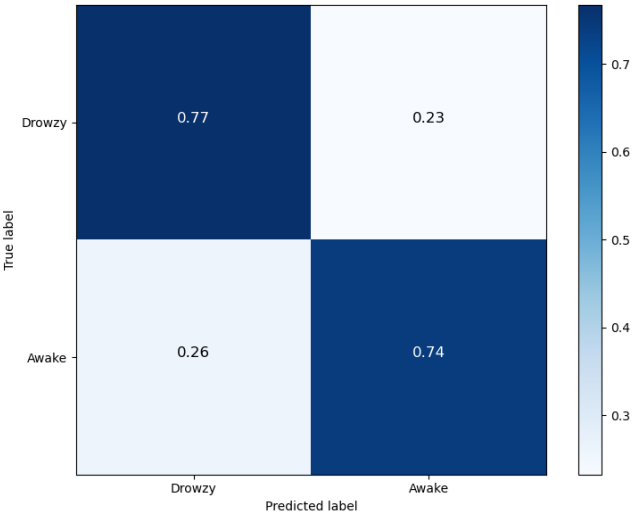

Figure 12: The confusion matrix of the 2D-1D CNN model in FL

accuracy results for each subject in the FL setup.

Table 9: The accuracy of drowsiness detection on test subjects' PVTs and their corresponding KSS scores

Subject	PVT	KSS	Accuracy
2	1	3	40.47%
3	1	2	100%
4	3	9	100%
7	3	9	84.52%
9	3	8	88.23%
11	3	7	9.52%
12	1	2	100%
14	3	8	100%

The global model's lower performance on unseen subjects, particularly subjects 2 and 11, whose accuracies dropped to 40.47% and 9.52% under FL despite achieving 100% in the centralized model, highlights a key challenge in FL, namely client drift. Since each client is trained only on its local data, the aggregated model may fail to capture patterns that generalize across all users, a problem that is amplified by the non-IID nature of data. Nevertheless, the FL model demonstrates overall robustness, achieving high accuracy for most unseen subjects. To better understand data heterogeneity and the performance degradation in FL, We generated t-SNE plots of feature vectors before FL training (Figure 13) and after (Figure 14).

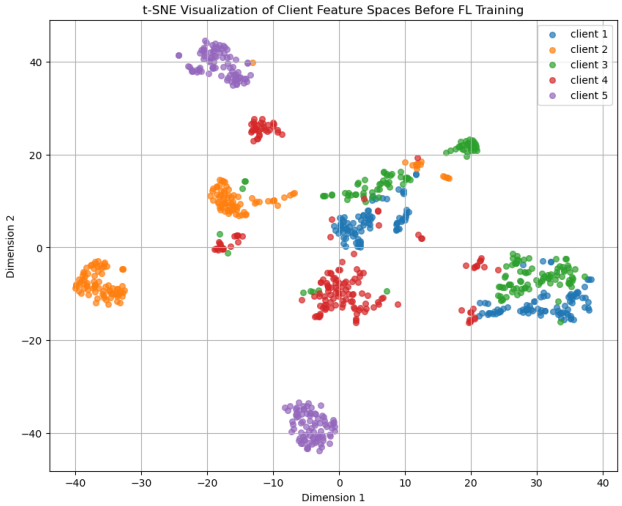

Figure 13: t-SNE visualization of the feature vectors before FL training

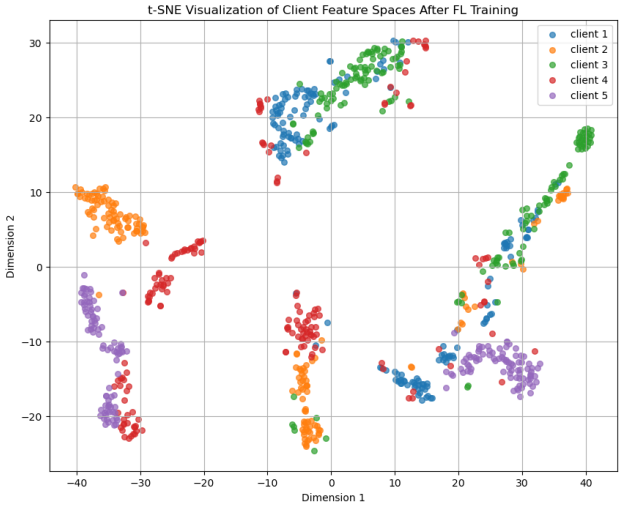

Figure 14: t-SNE visualization of the feature vectors after FL training

The t-SNE visualizations reveal that, prior to training, feature embeddings are unstructured and heavily client dependent. Following FL training, partial mixing between clients becomes apparent, and features organize into more coherent clusters, demonstrating that FedAvg effectively aligns representations across heterogeneous clients. Nevertheless, subject-level clustering persists, reflecting the non-IID nature of the data (one subject per client) and explaining the reduced generalization observed on unseen subjects.

We tested multiple local training rounds in order to see their effects on the global model accuracy. The results are shown in Table 10. The highest accuracy of 77.86% was

Table 10: Global model accuracy over local training rounds

Local round	Accuracy
5	77.86%
10	75.78%
15	73.99%
20	71.32%
25	75.33%

achieved with only 5 local training rounds, while accuracy decreased as the number of local rounds increased, with the exception of 25 rounds, which can be attributed to client drift and gradient noise. As each client trains on its local data, excessive local updates can diverge from the global optimum, temporarily reducing global model performance. These results suggest that minimal local training may be optimal for this drowsiness detection task, while excessive local training may affect the global model generalization.

5.6 Execution time

The execution time is crucial for detecting the driver drowsiness and preventing road traffic accidents and it must be done in real-time using low computational power and a low memory device. Therefore, we tested our system using the NVIDIA Jetson Nano module which is a small computer with NVIDIA Maxwell architecture with 128 NVIDIA CUDA® cores, Quad-core ARM Cortex-A57 MPCore processor, and 4 GB 64-bit LPDDR4, 1600MHz 25.6 GB/s, made for AI embedded application.

We achieved up to 25.6 FPS using ONNX Runtime, which exceeds the 15 FPS required by our system for efficient operation. In addition, it leaves sufficient FPS margin in case of frame drops that may occur in real-time applications. These results confirm the system's suitability for resource-constrained edge devices.

6 Discussion

Comparison with previous works

Compared with prior works discussed in Section 2, the proposed 2D–1D CNN achieves competitive or superior accuracy while remaining significantly lighter and more suitable for real-time edge deployment. Studies such as [21] and [22] rely on recurrent architectures (RNNs/LSTMs) or 3D-CNN [20] to capture temporal information. While these models are powerful, they are computationally heavier and

harder to deploy on constrained devices. Our use of a 1D-CNN for temporal modelling provides a more efficient alternative: it captures short-term temporal information while maintaining a small model, enabling real-time execution.
Furthermore, some previous studies evaluate their models using RGB datasets, controlled or uncontrolled environments that differ considerably from the NIR-based DROZY dataset. These differences in input data, subject variability, and labelling protocol make direct comparison challenging. Nevertheless, our results demonstrate that a lightweight spatio-temporal architecture can reach accuracy levels comparable to or exceeding those of larger models.

Sequence length analysis
Our sequence-length analysis shows performance peaks at a 7 second window, after which accuracy declines. From a modeling perspective, 1D-CNNs perform best when the temporal window contains sufficient but not excessive information. Very short sequences (1-3 seconds) do not capture enough temporal evolution of eye behavior, while very long sequences (15-20 seconds) introduce redundant or noisy information that dilutes the discriminative features. The 7-second window offers an optimal trade-off between temporal richness and noise, making it well-suited to the temporal capture capacity of the 1D-CNN.

Centralised and federating learning performance
The comparison between centralized learning and federated learning highlights important trade-offs between performance and privacy in the context of driver drowsiness detection. In the centralized learning, the model achieved 99% accuracy on the test set and 98.06% on unseen subjects. This approach benefits from full access to diverse and balanced data distributions, allowing the model to capture more generalized features and avoid client-specific biases, which explains its high performance.
In contrast, the federated learning model achieved 95.27% accuracy on the test set. While its results are lower than centralized learning, this performance is still strong and demonstrates the global model's ability to effectively learn from decentralized data across clients. However, the accuracy significantly dropped to 77.86% on unseen subjects' videos, which is substantially lower than the centralized model's performance. This drop in performance suggests client drift caused by training on non-IID data, a known challenge in FL when clients have non-IID data. Without direct access to a globally balanced dataset during training, certain patterns may be underrepresented, making it harder for the global model to generalize to new subjects.

Communication cost
The proposed model is lightweight. During each federated round, only the model weights are exchanged between clients and the server, resulting in minimal communication overhead compared to typical deep learning models. The overall communication cost grows linearly with the number of rounds and participating clients, but in our case it remains modest due to the compact model design.

Privacy leakage risks
Although federated learning prevents raw eye images from being shared, the gradients or model updates may still be vulnerable to refactoring attacks (reconstruction of the training datasets), membership inference attacks (inferring the identity of participants) or model extraction attacks (reconstruction of similar models). Various privacy-preserving mechanisms exist to mitigate these risks, such as cryptography-based federated learning protocols, differential privacy (adding random noise to hide the real information), and other privacy security techniques [56]. While our current implementation does not apply these mechanisms, they are complementary and could further strengthen the privacy-preserving capabilities of our framework.

Real life application of FL
The proposed lightweight FL based drowsiness detector have several real-world applications. In vehicles, the model can operate locally to trigger in-vehicle alerts when drowsiness is consistently detected. In fleet operations, federated learning enables vehicles to collaboratively improve the model while maintaining driver privacy. Furthermore, on-device processing is consistent with the European Data Protection Board's Guidelines 1/2020 on connected vehicles [57], which recommends the local processing of personal data and data minimization, as raw facial images are neither transmitted nor stored externally. These examples illustrate how the proposed architecture can be integrated into modern driver monitoring systems.

7 Conclusion

We presented a system based on CNN for detecting driver drowsiness that excels in both accuracy and FPS. The system was tested on the Jetson Nano, an edge device well-suited for embedded AI applications. Our proposed approach first detects the driver's face and then extracts the eye patches by estimating facial landmarks. These eye images are fed into the proposed 2D–1D CNN model to determine the driver's state.

We first proposed a 2D-CNN model trained on the MRL eye dataset to classify eye state into open or closed, which was then fine-tuned on the DROZY dataset. Using the same DROZY dataset, we trained a 1D-CNN model for video classification, achieving 99% accuracy on the test set and 98.06% on videos from subjects not included in the training set. To the best of our knowledge, these are the highest results reported on this dataset for two-class classification evaluated on unseen subjects, demonstrating that our system accurately determines driver drowsiness state and generalizes effectively to new, unseen data. Furthermore, the system achieves up to 25.6 FPS on a resource-constrained

device such as the Jetson Nano, which is higher than the 15 FPS frame rate used to process videos in this study. The results demonstrate the feasibility of deploying the proposed system on edge devices to monitor driver state.

In addition to traditional centralized learning, we evaluated the performance of our 2D-1D CNN model using a federated learning framework, where each subject was treated as a separate client. The aggregated global model achieved 95.27% accuracy on the evaluation set and 77.86% on unseen subjects' videos. This demonstrates the potential of FL to build privacy-preserving models while still achieving reasonable performance, though it highlights the challenge of generalizing from non-IID data. Future work could address this through adaptive mechanisms such as personalized federated learning or adaptive aggregation strategies, further enhancing the framework's applicability to real-world scenarios while maintaining privacy.

Despite the encouraging results, our 2D-1D CNN model relies solely on eye state to determine driver state, which can be misleading because eye closure patterns and how they reflect drowsiness differ among individuals. Monitoring other facial regions, such as the mouth, and physiological signals, including EEG and ECG could greatly improve the results. In addition, incorporating a dataset with varying lighting conditions, subjects wearing glasses, and different cameras would improve the robustness of the system. These limitations will be addressed in future work.

References

[1] World Health Organization. Road traffic injuries. https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries. Last accessed 2025-10-13.

[2] National Center for Statistics and Analysis. Drowsy driving 2015 (crash•stats brief statistical summary. Report no. dot hs 812 446), National Highway Traffic Safety Administration, Washington, DC, 10 2017.

[3] E Grandjean. Fatigue in industry. *Occupational and Environmental Medicine*, 36(3):175–186, 1979. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1136/oem.36.3.175.

[4] David F. Dinges. An overview of sleepiness and accidents. *Journal of Sleep Research*, 4(s2):4–14, 1995. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1111/j.1365-2869.1995.tb00220.x.

[5] Amina Guettas, Soheyb Ayad, and Okba Kazar. Driver state monitoring system: A review. In *Proceedings of the 4th International Conference on Big Data and Internet of Things*, 2020. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1145/3372938.3372966.

[6] Rustem Dautov and Erik Johannes Husom. Drowsiness detection using federated learning: Lessons learnt from dealing with non-iid data. In *Proceedings of the 17th International Conference on PErvasive Technologies Related to Assistive Environments*, 2024. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1145/3652037.3652074.

[7] Jianfeng Hu and Jianliang Min. Automated detection of driver fatigue based on eeg signals using gradient boosting decision tree model. *Cognitive neurodynamics*, 12(4):431–440, 2018. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1007/s11571-018-9485-1.

[8] Ali Amer Hayawi and Jumana Waleed. Driver's drowsiness monitoring and alarming auto-system based on eog signals. In *2019 2nd International Conference on Engineering Technology and its Applications (IICETA)*, 2019. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1109/IICETA47481.2019.9013000.

[9] Mohammad Mahmoodi and Ali Nahvi. Driver drowsiness detection based on classification of surface electromyography features in a driving simulator. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 233(4):395–406, 2019. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1177/0954411919831313.

[10] Rahul Bhardwaj, Priya Natrajan, and Venkatesh Balasubramanian. Study to determine the effectiveness of deep learning classifiers for ecg based driver fatigue classification. In *2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS)*, 2018. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1109/ICIINFS.2018.8721391.

[11] Muhammad Awais, Nasreen Badruddin, and Micheal Drieberg. A hybrid approach to detect driver drowsiness utilizing physiological signals to improve system performance and wearability. *Sensors*, 17(9):1991, 2017. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.3390/s17091991.

[12] Sazali Yaacob, Nur Afrina Izzati Affandi, Pranesh Krishnan, Amir Rasyadan, Muhyi Yaakop, and Firdaus Mohamed. Drowsiness detection using eeg and ecg signals. In *2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, 2020. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1109/IICAIET49801.2020.9257867.

[13] Sagara Sumathipala, Danodya Weerasinghe, Meleeshiya jayakody, Shashika Eranda, and Thanuja Gunasekara. Drowsiness detection and alert system for motorcyclist safety. In *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, 2020. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1109/ICTer51097.2020.9325467.

[14] Sadegh Arefnezhad, Sajjad Samiee, Arno Eichberger, Matthias Frühwirth, Clemens Kaufmann, and Emma Klotz. Applying deep neural networks for multi-level classification of driver drowsiness using vehicle-based measures. *Expert Systems with Applications*, 162:113778, 2020. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1016/j.eswa.2020.113778.

[15] Sadegh Arefnezhad, Sajjad Samiee, Arno Eichberger, and Ali Nahvi. Driver drowsiness detection based on steering wheel data applying adaptive neuro-fuzzy feature selection. *Sensors*, 19(4):943, 2019. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.3390/s19040943.

[16] Mohit Dua, Ritu Singla, Saumya Raj, and Arti Jangra. Deep cnn models-based ensemble approach to driver drowsiness detection. *Neural Computing and Applications*, 33(8):3155–3168, 2021. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1007/s00521-020-05209-7.

[17] Suhandi Junaedi and Habibullah Akbar. Driver drowsiness detection based on face feature and perclos. *Journal of Physics: Conference Series*, 1090(1):012037, 2018. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1088/1742-6596/1090/1/012037.

[18] Fouzia, R. Roopalakshmi, Jayantkumar A. Rathod, Ashwitha 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)*, 2018. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1109/ICICCT.2018.8473203.

[19] Caio Bezerra Souto Maior, Márcio José das Chagas Moura, João Mateus Marques Santana, and Isis Didier Lins. Real-time classification for autonomous drowsiness detection using eye aspect ratio. *Expert Systems with Applications*, 158:113505, 2020. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1016/j.eswa.2020.113505.

[20] Heming Yao, Wei Zhang, Rajesh Malhan, Jonathan Gryak, Najarian, and Kayvan. Filter-pruned 3d convolutional neural network for drowsiness detection. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2018. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1109/EMBC.2018.8512561.

[21] Shuang Chen, Zengcai Wang, and Wenxin Chen. Driver drowsiness estimation based on factorized bilinear feature fusion and a long-short-term recurrent convolutional network. *Information*, 12(1):3, 2021. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.3390/info12010003.

[22] Elena Magán, M. Paz Sesmero, Juan Manuel Alonso-Weber, and Araceli Sanchis. Driver drowsiness detection by applying deep learning techniques to sequences of images. *Applied Sciences*, 12(3):1145, 2022. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.3390/app12031145.

[23] Xiaofeng Li, Jiahao Xia, Libo Cao, Guanjun Zhang, and Xiexing Feng. Driver fatigue detection based on convolutional neural network and face alignment for edge computing device. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 235(10-11):2699–2711, 2021. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1177/0954407021999485.

[24] Hu He, Xiaoyong Zhang, Fu Jiang, Chenglong Wang, Yingze Yang, Weirong Liu, and Jun Peng. A real-time driver fatigue detection method based on two-stage convolutional neural network. *IFAC-PapersOnLine*, 53(2):15374–15379, 2020. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1016/j.ifacol.2020.12.2357.

[25] Shivam Khare, Sandeep Palakkal, TV Hari Krishnan, Chanwon Seo, Yehoon Kim, Sojung Yun, and Sankaranarayanan Parameswaran. Real-time driver drowsiness detection using deep learning and heterogeneous computing on embedded system. In *Computer Vision and Image Processing*, 2020. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1007/978-981-15-4018-9_8.

[26] William Lindskog, Valentin Spannagl, and Christian Prehofer. Federated learning for drowsiness detection in connected vehicles. In *Intelligent Transport Systems*, 2024. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1007/978-3-031-49379-9_9.

[27] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629v1*, 2016.

https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.48550/arXiv.1602.05629.

[28] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.48550/arXiv.1610.02527.

[29] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.48550/arXiv.1610.05492.

[30] Y. Supriya and Thippa Reddy Gadekallu. A survey on soft computing techniques for federated learning- applications, challenges and future directions. *ACM Journal of Data and Information Quality*, 15(2):1–28, 2023. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1145/3575810.

[31] Atiqa Zafar, Christian Prehofer, and Chih-Hong Cheng. Federated learning for driver status monitoring. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1109/ITSC48978.2021.9564936.

[32] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1109/MSP.2020.2975749.

[33] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10(60):1755–1758, 2009. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.5555/1577069.1755843.

[34] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *arXiv preprint arXiv:1905.00641*, 2019. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.48550/arXiv.1905.00641.

[35] Yuantao Feng, Shiqi Yu, Hanyang Peng, Yan-Ran Li, and Jianguo Zhang. Detect faces efficiently: A survey and evaluations. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 4(1):1–18, 2022. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1109/TBIOM.2021.3120412.

[36] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1109/CVPR.2001.990517.

[37] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *Proceedings of the British Machine Vision Conference*, 2014. https://doi.org/10.31449/inf.v50i12.12530http://dx.doi.org/10.5244/C.28.65.

[38] G. Gamage, I. Sudasingha, I. Perera, and D. Meedeniya. Reinstating dlib correlation human trackers under occlusions in human detection based tracking. In *2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer)*, 2018. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1109/ICTER.2018.8615551.

[39] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *2013 IEEE International Conference on Computer Vision Workshops*, 2013. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1109/ICCVW.2013.59.

[40] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1109/CVPR.2014.241.

[41] Hatem Sindi, Majid Nour, Muhyaddin Rawa, Şaban Öztürk, and Kemal Polat. Random fully connected layered 1d cnn for solving the z-bus loss allocation problem. *Measurement*, 171:108794, 2021. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1016/j.measurement.2020.108794.

[42] Yuki Watanabe, Taiki Kimura, Tetsuaki Matsunawa, and Shigeki Nojima. Accurate lithography simulation model based on convolutional neural networks. In *Optical Microlithography XXX*, 2017. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1117/12.2257871.

[43] Kui Liu, Guixia Kang, Ningbo Zhang, and Beibei Hou. Breast cancer classification based on fully-connected layer first convolutional neural networks. *IEEE Access*, 6:23722–23732, 2018. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1109/ACCESS.2018.2817593.

[44] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman. 1d convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, 151:107398, 2021. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1016/j.ymssp.2020.107398.

[45] Shamsi Shekari Soleimanloo, Vanessa E. Wilkinson, Jennifer M. Cori, Justine Westlake, Bronwyn Stevens, Luke A. Downey, Brook A. Shiferaw, Shantha M. W. Rajaratnam, and Mark E. Howard. Eye-blink parameters detect on-road track-driving impairment following severe sleep deprivation. *Journal of Clinical Sleep Medicine*, 15(09):1271–1284, 2019. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.5664/jcsm.7918.

[46] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.48550/arXiv.1312.4400.

[47] Radovan Fusek. Pupil localization using geodesic distance. In *International Symposium on Visual Computing*, 2018. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1007/978-3-030-03801-4_38.

[48] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.48550/arXiv.1412.6980.

[49] Yangfan Zhou, Xin Wang, Mingchuan Zhang, Junlong Zhu, Ruijuan Zheng, and Qingtao Wu. Mpce: A maximum probability based cross entropy loss function for neural network classification. *IEEE Access*, 7:146331–146341, 2019. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1109/ACCESS.2019.2946264.

[50] Quentin Massoz, Thomas Langohr, Clémentine François, and Jacques G. Verly. The ulg multimodality drowsiness database (called drozy) and examples of use. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1109/WACV.2016.7477715.

[51] Torbjörn Åkerstedt and Mats Gillberg. Subjective and objective sleepiness in the active individual. *International Journal of Neuroscience*, 52(1-2):29–37, 1990. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.3109/00207459008994241.

[52] Torbjörn Åkerstedt, Anna Anund, John Axelsson, and Göran Kecklund. Subjective sleepiness is a sensitive indicator of insufficient sleep and impaired waking function. *Journal of Sleep Research*, 23(3):242–254, 2014. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1111/jsr.12158.

[53] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.48550/arXiv.1409.1556.

[54] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACMs*, 60(6):84–90, 2017. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1145/3065386.

[55] Miguel García-García, Alice Caplier, and Michèle Rombaut. Sleep deprivation detection for real-time driver monitoring using deep learning. In *International conference image analysis and recognition*, 2018. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1007/978-3-319-93000-8_49.

[56] Lina Ge, Haiao Li, Xiao Wang, and Zhe Wang. A review of secure federated learning: Privacy leakage threats, protection technologies, challenges and future directions. *Neurocomputing*, 561:126897, 2023. https://doi.org/10.31449/inf.v50i12.12530https://doi.org/10.1016/j.neucom.2023.126897.

[57] European Data Protection Board. Guidelines 1/2020 on processing personal data in the context of connected vehicles and mobility related applications. https://www.edpb.europa.eu/system/files/2021-03/edpb_guidelines_202001_connected_vehicles_v2.0_adopted_en.pdf, March 2021.