sciendo

Journal of Electrical Engineering, Vol. 76, No. 4, 2025, pp. 344-355

Modular deep residual network for driver stress detection based on photoplethysmography

Stevica Cvetkovic, Sandra Stankovic, Goran Stancic

Conventional approaches to driver stress detection frequently depend on complex multimodal sensor systems, which involve intrusive and costly data acquisition procedures. In contrast, this study explores the hypothesis that the photoplethysmography (PPG) signal can independently provide reliable indicators of stress. To evaluate this premise, we present a modular end-to-end deep residual network specifically designed to balance high classification accuracy with relatively low computational complexity. The architecture incorporates several optimization techniques, starting with optimal design of input layers that reduce spatial resolution and extract key features, eliminating the need for manual feature engineering. By leveraging bottleneck residual blocks enhanced with multi-branch mechanisms, our model improves overall accuracy. Each branch within the block applies distinct convolutional operations, enabling the extraction of both global and local features across multiple hierarchical levels. A comprehensive evaluation on two public datasets demonstrates that our model achieves accuracy over 98%, outperforming state-of-the-art methods while balancing complexity and performance. These findings highlight the model's potential for integration into real-world driving environments.

Keywords: driver stress recognition, deep learning, residual neural network, photoplethysmography, intelligent vehicles

1 Introduction

Detection of driver stress is critical to many applications, particularly in the development of advanced driver assistance systems (ADAS) and autonomous vehicles. Stress detection can help improve driver safety by identifying situations where a driver may be impaired due to stress, enabling timely intervention. Furthermore, understanding stress levels can assist in enhancing the overall driving experience by personalizing the in-vehicle environment, such as adjusting seat comfort, temperature, or audio settings based on the driver's emotional state. Extensive research on stress monitoring using wearable devices has been conducted in [1]. Several sensor modalities, including electrodermal activity (EDA), skin temperature (TEMP), accelerometery (ACC), electromyography (EMG), and electrocardiogram (ECG), have been explored for monitoring driver stress [2]. However, this paper focuses on photoplethysmography (PPG) as the sole modality for stress detection. The PPG signal provides valuable insight into dynamic blood volume pulse (BVP) changes in the body's extremities, offering a direct measurement of several physiological and cardiovascular parameters, including heart rate (HR), heart rate variability (HRV), oxygen saturation (SpO2), and respiratory rate (RESP) [3, 4]. In stressful situations, fluctuations in HR and HRV, driven by autonomic nervous system activity, lead to alterations in HR and HRV patterns, making them effective indicators of stress levels in drivers [5].

While multi-modal sensor systems may offer advantages in certain contexts [6], this study focuses exclusively on PPG sensors for driver stress detection for several compelling reasons. First, PPG is a non-invasive optical technique that measures blood volume changes through the skin using simple sensors [7]. This approach eliminates the need for direct skin contact or electrodes, minimizing the risk of irritation and enhancing user comfort. Additionally, PPG can be seamlessly integrated into commonly used wearable devices, such as smartwatches or fitness trackers, making it more acceptable and comfortable for drivers. This integration allows them to maintain regular activities without added discomfort. Moreover, PPG provides crucial physiological metrics, including heart rate, heart rate variability, and arterial stiffness – parameters strongly correlated with stress levels. These features serve as reliable indicators of stress, eliminating the need for additional sensors. Finally, by relying solely on PPG, the cost of sensor deployment and maintenance is significantly reduced compared to multi-modal setups, making it a cost-effective solution for stress detection in automotive applications.

Examples of signal segments corresponding to three stress level categories (low, medium, and high) are provided in Fig. 1. It is evident from the visual representations that the recorded PPG signal contains a considerable amount of noise, presenting a significant challenge for traditional machine learning techniques that rely on manual feature extraction. However, recent advancements in deep learning, which have demonstrated remarkable success in fields such as computer vision [8] and natural language processing [9], have encouraged researchers to investigate its potential for

Faculty of Electronic Engineering, University of Nis, 18000 Nis, Serbia
stevica.cvetkovic@elfak.ni.ac.rs

https://doi.org/10.2478/jee-2025-0035, Print (till 2015) ISSN 1335-3632, On-line ISSN 1339-309X

addressing complex time-series classification problems. A key advantage of deep learning methods lies in their inherent ability to automatically extract novel discriminative features directly from noisy input signals.

Motivated by this potential, our goal was to develop a highly accurate deep learning model for driver stress recognition using only PPG signals, while ensuring the network remains compact and executes rapidly. Achieving this objective requires the development of a modular end-to-end deep learning architecture, denoted as ***PPGResNet***, that minimizes operational complexity without compromising accuracy. This architecture prioritizes both flexibility and performance, enabling effective analyses of raw, non-preprocessed signals, which is essential for real-time applications where computational resources may be limited. The proposed network's design is meant to serve as a baseline and can be further simplified by downsampling or filtering the input signal prior to feeding it into the model.

Thus, the main contributions of this study are summarized as follows:

- We propose a compact, modular end-to-end deep learning model for accurate driver stress detection using only PPG signals. Efficiency is achieved through input layers that reduce spatial resolution and extract key features, followed by bottleneck residual blocks with a multi-branch design.
- The proposed network design can serve as a baseline and can be further simplified by downsampling or filtering the input signal. Its modular structure allows for easy adjustments and extensions, enabling the model to be adapted to various biomedical signals without the need for a complete redesign.
- We analyze design trade-offs and provide a quantitative evaluation of accuracy and efficiency using two public driver stress datasets. Our model achieves over 98% accuracy on both, demonstrating strong performance.

The paper is organized as follows: Section 2 presents an overview of the current literature on the application of artificial intelligence methods for assessing driver stress levels. Section 3 describes the proposed multi-branch deep residual network architecture. Section 4 details the benchmark datasets, data preprocessing procedures, and implementation specifics. The achieved results are presented and discussed in Section 5. Finally, Section 6 summarizes the conclusions of the paper and outlines directions for future research.

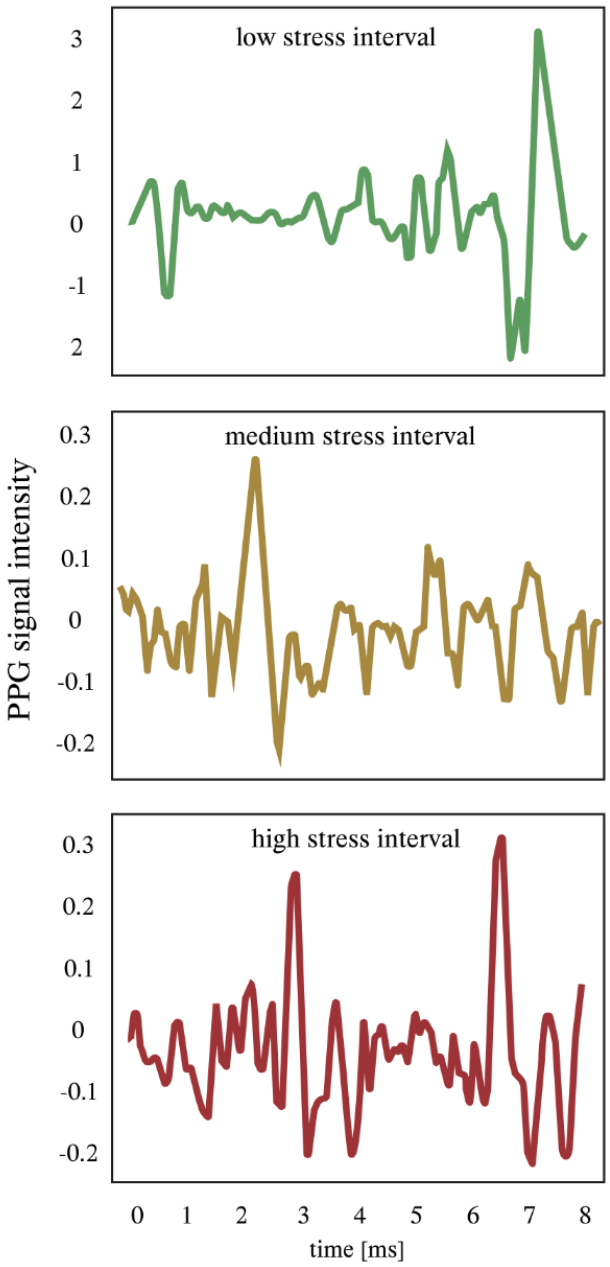

Fig. 1. Examples of PPG signals during low, medium, and high stress intervals

2 Related work

Earlier time-series classification approaches primarily relied on traditional machine learning methods such as Decision Trees, Support Vector Machines (SVM), and k-Nearest Neighbors (kNN) [10, 11]. Feature extraction can be performed across various domains using linear and non-linear methods or by transforming, combining, or aggregating existing features [2, 12].

In [13], driver stress classification was conducted by extracting statistical features and utilizing a Random Forest algorithm for classification. Similarly, the study in [14] aimed to differentiate between low and high stress levels by grouping drivers into different profiles based on physiological responses derived from EDA and HR. A normalized spectral clustering technique was employed for profile clustering. The impact of varying window sizes was explored in [15], revealing that longer windows enhance classification performance. However, the use of shorter window sizes may be preferable in dynamic environments, such as driving, where rapid changes require a more responsive stress detection approach.

Deep learning offers an alternative approach in which models autonomously extract features from raw time-series data, thereby eliminating the need for manual feature engineering. However, a known limitation of deep learning methods is their relatively high computational cost, particularly when employing deep network architectures and processing large input datasets. This challenge motivated our research to investigate the optimal design of a deep learning model that balances computational efficiency with high classification accuracy for time-series data. When examining driver stress levels, the deep learning architecture in [16] consists of several convolution layers with varying numbers of filters followed by a Global Average Pooling layer, a FC layer, and an LSTM layer. Another method that was employed is a Self-Supervised Learning approach [17]. Here, an encoder was trained using distinct representations of the following signals: ACC, BVP, EDA and TEMP. The architecture was inspired by Inception model [18], featuring blocks with multiple parallel layers whose outputs are stacked to form the final output. Multi-branch architectures were utilized in [19] by incorporating separate branches for each signal and a concatenating layer to learn shared encoded features. The overall loss was defined as the sum of losses from all branches. The features were extracted from BPV, EDA and TEMP signals, resulting in a total of 72 features. In [20] relatively high stress detection accuracy was achieved by fusing EDA, EMG, RESP, and TEMP signals. Each signal was processed through a dedicated convolutional block, and their outputs were concatenated and passed through fully connected layers in a three-stage sequence.

Recurrent Neural Networks (RNNs) were preferred options for time-series problems in the past [21, 22], due to their ability to handle sequences of varying lengths [23, 24]. They incorporate the concept of a hidden state within each unit, serving as the memory of the unit. However, learning long-term dependencies with RNNs poses challenges as gradients propagated across many stages tend to vanish or explode [24]. To address these challenges, gated RNNs are employed [25], the most common ones being Long Short-Term Memory (LSTM) networks [9, 26]. In multimodal wearable activity recognition, architectures like *DeepConvLSTM* are widely used [27]. Another influential deep learning architecture, the Residual Neural Network (ResNet), was originally introduced in [28], it reformulates the learning objective by enabling the network to learn a residual function defined as $F(x) = H(x) - x$, rather than directly approximating the target mapping $H(x)$. This transformation allows the original function to be expressed as $F(x) + x$, which is implemented using shortcut connections in a feedforward neural network. These connections bypass one or more layers without introducing additional parameters or computational overhead.

To support stable and efficient training in deep convolutional neural networks (CNNs), residual connections are employed to alleviate the vanishing gradient problem, thereby enabling the network to effectively learn complex patterns in time-series data such as PPG signals. Although CNNs are not inherently designed to capture long-range sequential dependencies, this limitation is not limited to critical for our application. The PPG signal is characterized by a quasi-periodic structure linked to individual heartbeats, with each cycle typically lasting around one second [29]. Moreover, wearable PPG sensors generally operate at relatively low sampling rates, typically ranging from 50 to 200 Hz.

To further enhance model performance, we introduce multi-branch Residual Blocks that process multiple temporal resolutions in parallel, enabling the extraction of both short-term and long-term features [30].

3 Proposed modular deep residual network architecture

In this section, we introduce the *PPGResNet*, a modular deep residual network architecture specifically designed for classifying PPG time-series data. It is organized into three primary stages: an optimized input module, a central core composed of multi-branch residual blocks, and a final output module. The input stage employs several layers designed to reduce spatial resolution efficiently while highlighting the most salient features. This is followed by a sequence of *Multi-Branch Residual Blocks* (MBRBs), which offer modularity by allowing easy adjustment of both the number and configuration of blocks without needing to redesign the overall architecture. By combining outputs from multiple parallel branches, the model effectively captures both global and local temporal dynamics. A key aspect of the design is the integration of bottleneck layers within the residual blocks, which substantially decrease computational demands while maintaining high accuracy. The network concludes with output layers that incorporate regularization techniques and enable the modelling of complex, long-range depen-

dencies. Figure 2 provides a comprehensive overview of the full network architecture, with further details described in the subsequent sections.

3.1 Input layers

Our input layer structure prioritizes two key aspects: reducing spatial resolution and selecting prominent signal features [28]. The model architecture begins with a strided convolutional layer with a larger filter size, commonly used for *downsampling* [31]. This is followed by a Max Pooling layer, which further reduces spatial dimensionality and enhances feature selection by retaining the maximum value within each pooling region, effectively filtering out less relevant and noisy information.

Let the input signal be represented as tensor $X \in \mathbb{R}^{TxC}$, where is the number of time steps and $C = 1$ is the number of input channels, as only a single wrist sensor signal is used as prior studies indicate minimal variation in PPG signals between the wrists [32]. Each *convolutional layer* applies F_i filters with kernel size k_i, stride s_i, and padding P_i, resulting in a temporal dimension reduction described by:

$$T_{i+1} = \left[\frac{T_i - k_i + 2P_i}{s_i} + 1\right] \quad (1)$$

while the number of channels expands from C to F_i, after the i[th] convolution. Similarly, the *Max Pooling* layer reduces the temporal dimension by a factor equal to its window size p:

$$T_{j+1} = \left[\frac{T_j}{p}\right] \quad (2)$$

Through this sequence of operations, the temporal resolution is progressively reduced, balancing computational efficiency with preservation of critical features. All convolutional layers have 64 output channels, offering an effective trade-off between model complexity and performance.

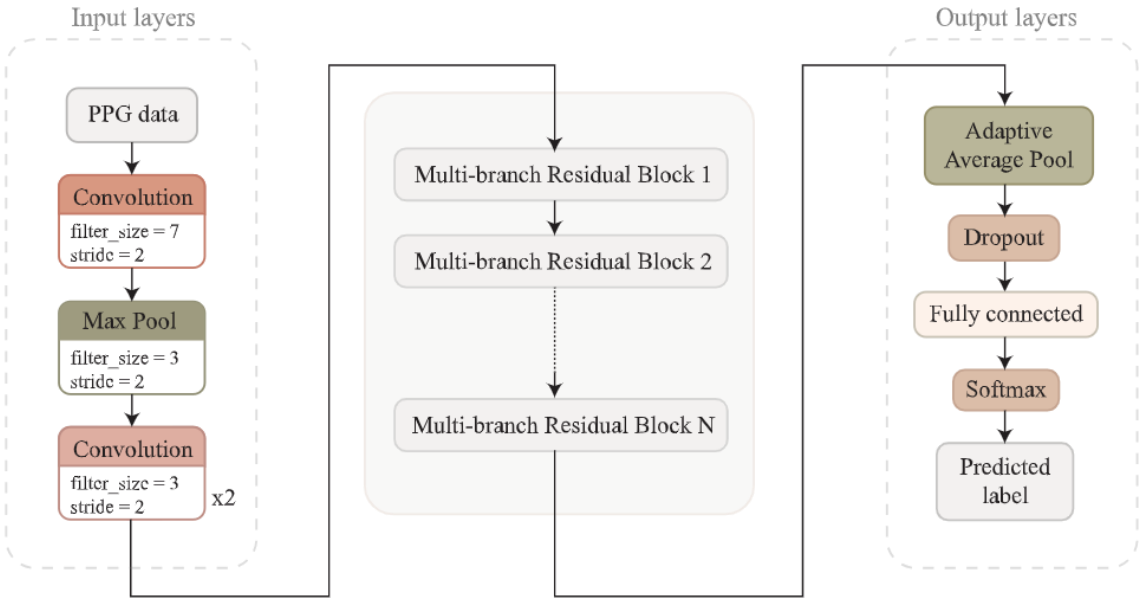

Fig. 2. The proposed *PPGResNet* – a modular residual network architecture with stacked MBRBs

3.2 Multi-branch residual blocks (MBRBs)

After the initial input data processing, the model incorporates a series of N stacked *MBRBs*, as illustrated in Fig. 3. Each block consists of multiple parallel branches, allowing the network to extract multi-scale and diverse features from the input. The blocks can vary in structure, offering flexibility for feature extraction at different depths. Using multiple branches instead of increasing depth helps shorten the paths between input and output, improving the model's capacity to learn long-range dependencies [33].

Formally, given the input feature map $X \in \mathbb{R}^{CxL}$ with C channels and a sequence length L, each MBRB computes its output Y as:

$$Y = X + \sum_{i=1}^{m} \mathcal{F}_i(X) \quad (3)$$

where m is the number of parallel branches, and $\mathcal{F}_i(\cdot)$ is the nonlinear transformation performed by the i^{th} branch. For each branch, $\mathcal{F}_i$ is modelled as a sequence of convolutional layers combined with Batch Normalization and activation functions. For a branch with k layers:

$$\mathcal{F}_i(X) = Conv_{i,k}\left(Conv_{i,k-1}\left(\dots Conv_{i,1}(X)\right)\right) \quad (4)$$

It is important to note that feature maps X and $\mathcal{F}_i(X)$ have compatible dimensions for element-wise addition. This multi-branch structure enables each branch to capture features at different temporal resolutions, using varying filter sizes per branch. The residual connection X allows gradients to flow more easily during back-propagation, mitigating vanishing gradient problems and improving training stability.

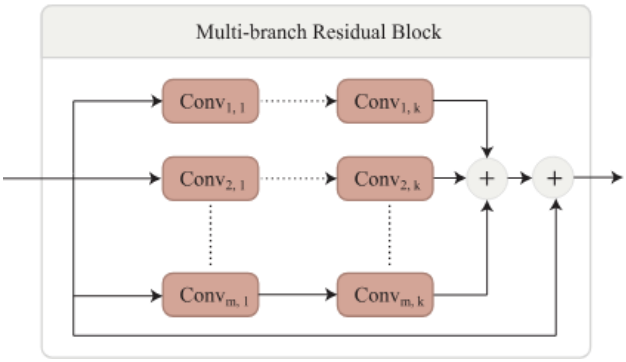

Fig. 3. Generic representation of the MBRB

For practical implementation, we explored several variations of these blocks:

- **Single-branch residual block:** This is the simplest configuration consisting of a single branch with two convolutional layers and a residual shortcut, similar to the original *ResNet* basic block [28]. We selected a filter size of 3 to capture fine-grained local features, assuming that global features are already abstracted by preceding input layers. Larger filters could capture broader contexts, but they also increase parameter count and risk overfitting.
- **3-branch residual block:** This configuration has three parallel branches with different filter sizes, enabling the block to extract multi-scale features simultaneously. Even if branches share the same filter size, their independent initializations allow them to learn distinct features. This promotes richer feature representations and network stability by providing multiple signal propagation paths.
- **3-branch residual block with bottleneck:** Inspired by the bottleneck design introduced in the original *ResNet* architecture [18, 30], this variant introduces 1×1 convolutions as linear projections to reduce the number of parameters and computational cost. The bottleneck module first compresses feature dimensionality via a 1×1 convolution, then applies larger kernel convolutions to extract complex patterns, followed by another 1×1 convolution to restore the original channel dimensions.

Each convolutional layer is followed by a Batch Normalization (BN) layer, which addresses issues such as exploding and vanishing gradients and helps prevent the network from becoming stuck in suboptimal minima [18]. By normalizing the inputs to each layer, BN stabilizes the learning process and accelerates convergence. Furthermore, it has a slight regularization effect that can help reduce the risk of overfitting [8].

To enable the network to model complex, non-linear relationships, we apply the *Rectified Linear Unit* (ReLU) activation function after each normalization step.

3.3 Output layers

Before reaching the final layer, we incorporate an Adaptive Average Pooling layer followed by a Drop-out layer and a Fully Connected classification layer.

The Adaptive Average Pooling layer compresses each feature map to a fixed-size output, regardless of the original temporal length of the signal. This ensures consistent input dimensions for the following layers and improves generalization by summarizing temporal features:

$$z_c = \frac{1}{L}\sum_{i=1}^{L} x_{c,t} \quad (5)$$

where $x_{c,t}$ is the activation at time step t and channel c, and L is the temporal dimension after convolution and pooling.

To further improve generalization and prevent overfitting, we apply Dropout with probability $p = 0.2$, which randomly disables the fraction of the feature elements during training. The resulting feature vector is then passed to the Fully Connected Layer that computes class logits. These are transformed into output probabilities using the *Softmax* function.

4 Evaluation details

To conduct a comprehensive evaluation of driver stress level recognition, we employed two publicly available datasets.

4.1 Datasets

The **AffectiveROAD dataset** [37] contains 13 driving sessions performed by 10 distinct drivers. Each drive spanned approximately 85 minutes, incorporating a 30-minute rest interval. These routes include a variety of road types and environmental conditions, leading to varying degrees of stress. A hu-man observer, situated in the back seat, subjectively estimated the stress levels using a laptop-based slider scale that ranged from 0 (no stress) to 1 (extremely high stress). After the drives, the drivers were asked to review and, if needed, change the stress scores based on their perception. Participants wore two physiological devices: the Empatica E4 on both wrists and the Zephyr Bioharness 3 on the chest, capturing data like BVP, EDA, TEMP, and ACC.

Additional evaluation was conducted using the multimodal dataset for Wearable Stress and Affect Detection – WESAD [38]. The data was collected using two recording devices: the RespiBAN Professional on the subject's chest and the Empatica E4 on the wrist of the non-dominant hand. A total of 17 subjects participated in the study, but the data from two participants was excluded due to sensor mal-function. Participants first had a 20-minute baseline session, followed by 11 video clips to induce amusement, then the Trier Social Stress Test including a speech and mental arithmetic. A guided meditation followed to return them to baseline. Both datasets used the Empatica E4 sensor to record BVP data at 64 Hz.

4.2 Data preprocessing

To capture local temporal characteristics and dynamics, it is common to segment the continuous time series into smaller overlapping or non-overlapping segments, known as sliding windows or time windows. Before segmenting the data into windows, all features were standardized to ensure comparable scales across variables and improve model training stability. This preprocessing step reduces bias caused by differing magnitudes of sensor measurements and facilitates faster and more reliable convergence during model training.

Sliding window segmentation enables the analysis of short segments of the signal, preserving temporal resolution while increasing the amount of data for feature extraction and model training. This approach is particularly useful for handling non-stationary signals, where statistical properties change over time, as it allows the model to focus on local temporal contexts. The choice of window length and the amount of overlap between consecutive windows are crucial parameters. Larger windows may capture more global information but can smooth over important transient features, while smaller windows emphasize local details but might lose context. Overlapping windows help to maintain continuity between segments and provide a more comprehensive representation of the time series.

Each window has a fixed length W, and con-secutive windows overlap by O samples. To quantify the overlap, we define the overlap ratio:

$$\alpha = \frac{O}{W}, 0 \leq \alpha < 1 \tag{6}$$

which represents the fraction of the window length shared between adjacent windows. The starting index of the ith window, denoted s_i, is given by:

$$s_i = i\,W\,(1-\alpha), i = 0, 1, 2, \ldots \tag{7}$$

which shifts each window by a fraction $1-\alpha$ of its length to ensure consistent overlap. Thus, the ith window covers the following data segment:

$$window_i = [s_i, s_i + W - 1] \tag{8}$$

To effectively apply sliding window segmentation on these datasets, it is crucial to assign consistent labels to each window segment. The **WESAD** dataset provides ground truth labels for four distinct affective states (*baseline, stress, amusement, and meditation*) at each time step. When segmenting the data into windows, some windows may contain conflicting labels; in such cases, the label assigned to the window is determined by majority voting, selecting the affective state that appears most frequently within the window. On the other hand, the **AffectiveROAD** dataset provides continuous relative stress scores rather than discrete labels. To utilize these scores for classification, we compute the average stress level within each window and discretize it into three categories: *low* (0 – 0.4), *medium* (0.4 – 0.75), and *high* (0.75 – 1), following the methodology in [39]. This categorization is based on the assumption that stress levels are generally high during city driving, medium on highways, and low during rest periods.

4.3 Implementation details

The models were developed and trained on virtual machines equipped with *Intel Xeon CPUs* running at 2.2 GHz and 12 GB of RAM, providing a balanced computational environment for efficient experimentation. Our implementations were carried out using the *PyTorch* deep learning framework, which offers flexibility and optimized performance for training neural networks. For optimization, we employed the Adam optimizer due to its adaptive learning rate capabilities,

setting an initial learning rate of 0.002. Training was performed over 30 epochs with a learning rate scheduler that reduced the rate by 85% every 3 epochs to ensure stable convergence. Model performance was assessed using 5-fold cross-validation for robustness across data splits. To ensure reproducibility, a fixed random seed controlled all sources of randomness, including data shuffling and weight initialization.

5 Experimental results

For the experimental evaluation, we will use the following notation to represent specific configurations of our modular network:

$$PPGResNet\boldsymbol{N}\ (s_1\ -s_2\ -\ldots-s_k\ ,[bottleneck])$$

where N represents the number of *MBRBs*, values s_i represent the filter sizes of the filters in each of the branches. The optional *[bottleneck]* tag indicates the use of a bottleneck structure within the block. As a first step in our evaluation, we sought to determine the optimal sliding window length by comparing model accuracy across a range of window sizes, from 4 to 16 seconds. The overlap between consecutive windows was fixed at 75%. The results for the AffectiveROAD dataset are presented in Table 1.

For the evaluation, we use well-established metrics: accuracy and F1 score.

Accuracy *(Acc.)* is defined as the ratio of correctly classified samples to the total number of samples:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

The **F1 score** is the harmonic mean of precision and recall, providing a balanced measure especially in the presence of class imbalance:

$$F1\ score = 2\ \frac{Precision\ \cdot Recall}{Precision + Recall} \tag{10}$$

where:

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN} \tag{11}$$

Here, TP, TN, FP, and FN refer to true positives, true negatives, false positives, and false negatives, respectively.

Based on the results obtained, we selected an 8-second window size for further experiments and proceeded to assess the effects of different overlap ratios between consecutive sliding windows. The impact on classification accuracy is summarized in Tab. 2, where we compared overlap ratios of 50%, 75%, and 87.5%. These specific overlap values were chosen because all windows in the 50% overlap set are also included in the 75% overlap set, and the same holds true for the 75% and 87.5% overlap sets. This approach effectively increases the sample size while retaining previous samples, thereby enlarging the training dataset and reducing the risk of overfitting. Although the training accuracies across all three overlap variations are quite similar, the validation accuracies highlight the differences in performance.

Table 1. Comparison of different window lengths on the AffectiveROAD Dataset (75% overlap)

	4 seconds		**8 seconds**		**12 seconds**		**16 seconds**	
Model	**Acc.**	**F1**	**Acc.**	**F1**	**Acc.**	**F1**	**Acc.**	**F1**
PPGResNet_1 (3)	75.1	69.73	81.53	77.29	78.10	74.36	76.89	70.03
PPGResNet_3 (3)	85.50	83.54	89.87	88.83	92.42	90.89	84.45	83.07
PPGResNet_5 (3)	85.61	82.65	91.02	89.49	91.87	90.59	88.52	86.49

Table 2. Comparison of overlap ratios in the AffectiveROAD Dataset with a fixed 8s window length

	50% overlap		**75% overlap**		**87.5% overlap**	
Model	**Acc.**	**F1**	**Acc.**	**F1**	**Acc.**	**F1**
PPGResNet_1 (3)	59.22	54.48	81.53	77.29	81.10	83.71
PPGResNet_3 (3)	71.30	60.93	89.87	88.83	97.93	97.64
PPGResNet_5 (3)	73.60	67.53	91.02	89.49	97.73	97.44

5.1 Variations of the MBRB

Following the initial experiments, we identified PPGResNet_3 as the most effective configuration. It consistently outperformed the single-block model and achieved performance comparable to the deeper five-block variant, making it a favorable balance between complexity and accuracy. In the next phase of evaluation, we investigated architectural variations of this three-block model by modifying the internal structure of the residual blocks. These variations, described previously in Section 3, involved multi-branch designs with various filter size combinations: 1-3-5, 3-5-7, and 3-5.

Our primary evaluation was conducted using 8-second windows with an 87.5% overlap, which had previously yielded the best results. We also evaluated the same architectures with a reduced 75% overlap, as it reveals more pronounced differences in the results. Both the AffectiveROAD and WESAD datasets were used in this analysis, and the corresponding results are presented in Table 3.

We observe that adding more branches to the model generally enables it to learn more meaningful and diverse features. When comparing the (3-5-7) and (1-3-5) architectures, we find that smaller kernel sizes tend to yield better results. This can likely be attributed to the fact that the input vector has already been downsampled by the initial layers, allowing the smaller kernels in subsequent layers to capture finer, more detailed features in the data. These finer details seem to be more critical for achieving higher performance in this task.

When we remove the 1×1 convolutional branch from the (1-3-5) architecture, resulting in the (3-5) configuration, there is a noticeable drop in accuracy. This implies that, although these 1×1 convolutions are conceptually simple – performing pointwise multiplications – they play a crucial role in enhancing feature extraction by enabling the model to learn more complex combinations of features across different channels.

Furthermore, adding a bottleneck layer does not significantly reduce the model's accuracy, but it does help in reducing model complexity. By compressing the feature space and forcing the model to focus on the most important features, the bottleneck layer reduces the number of parameters, leading to a more efficient model without a major sacrifice in performance. As we will explore further, this reduction in complexity allows for faster training and inference, making the model more suitable for real-time or resource-constrained applications.

Table 3. Comparison of model variations at different overlap ratio

	AffectiveROAD dataset				**WESAD dataset**			
	75% overlap		**87.5% overlap**		**75% overlap**		**87.5% overlap**	
Model	**Acc.**	**F1**	**Acc.**	**F1**	**Acc.**	**F1**	**Acc.**	**F1**
PPGResNet_3 (3)	89.87	88.32	97.93	97.64	96.25	95.68	99.35	99.26
PPGResNet_3 (1-3-5)	93.47	92.57	98.33	98.10	97.59	97.27	99.66	99.60
PPGResNet_3 (3-5-7)	91.34	89.96	89.06	97.77	97.44	97.06	99.55	99.49
PPGResNet_3 (3-5)	92.48	91.43	98.24	98.01	97.05	96.66	99.59	99.53
PPGResNet_3 (1-3-5 bottleneck)	9174	90.58	98.18	97.95	97.05	96.09	99.48	99.40

Table 4. Comparison with state-of-the-art models from the literature, using two public datasets

Dataset	Method	Acc. (%)	Signals
AffectiveROAD dataset	Random Forest [14]	70.40	PPG
	SVM-RBF [14]	83.00	EDA + PPG
	DNN [16]	87.95	PPG + ACC + EDA + TEMP + ECG + RESP
	PPGResNet_3 (1-3-5)	98.33	PPG
WESAD dataset	Self-supervised DNN [17]	87.13	PPG + ACC + EDA + TEMP
	Random Forest [15]	81.40	PPG
	Ensemble of DNNs [19]	89.94	PPG
	DNN [20]	93.64	ECG + EDA + EMG + RESP + TEMP
	PPGResNet_3 (1-3-5)	99.66	PPG

5.2 Comparison with state-of-the-art results

To evaluate the performance of our optimal model configuration, we compared it with several state-of-the-art approaches from the literature, focusing primarily on classification accuracy. Table 4 summarizes the results of these comparisons, highlighting methods that either utilize deep neural networks (DNN) or rely solely on PPG signals. Figure 4 presents a visual comparison where the accuracies are compared, and different colors correspond to different number of input signals. It is evident that our proposed *PPGResNet* model consistently outperforms other methods, even those that incorporate multiple physiological signals.

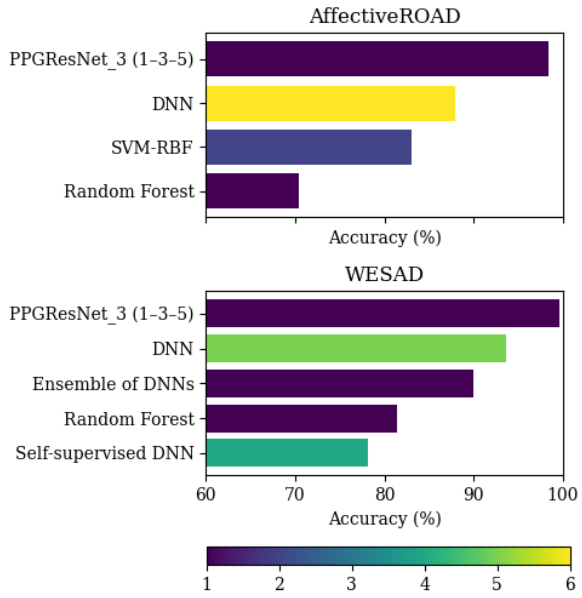

Fig. 4. Comparison of the performance of our *PPGResNet* alongside state-of-the-art models. Different colors correspond to different number of input signal types.

For the AffectiveROAD dataset, we see that simpler models like Random Forest, which uses only PPG signals, achieve a modest accuracy of 70.40%. This underscores the inherent difficulty of working with a single signal, where more advanced models and multi-signal approaches are often necessary to capture the nuances of physiological data. In contrast, methods that combine multiple signals, such as SVM-RBG and DNN, show marked improvements in accuracy. While these methods benefit from the extra data provided by multiple signals, they also come with their own set of challenges, including the need for specialized sensors and more complex data processing pipelines. In comparison, our *PPGResNet_3* (1-3-5) model, which uses only PPG signals, achieves an outstanding 98.33% accuracy. This demonstrates the power of our deep learning architecture to extract meaningful features from a single signal, making it a highly efficient and practical solution for stress detection tasks.

Similarly, on the WESAD dataset, the Self-supervised DNN method, which relies on PPG, ACC, EDA, and TEMP signals, achieves 87.13% accuracy. Again, incorporating additional signals leads to improved performance, but at the cost of increased complexity and sensor requirements. On the other hand, our model *PPGResNet_3* (1-3-5), which uses only PPG signals, outperforms all other methods, reaching 99.66% accuracy.

The results from both datasets highlight a crucial observation: while multi-signal methods can improve accuracy, they come with trade-offs. Larger input vectors, necessitated by the use of more signals, can lead to slower processing times and higher computational demands. Moreover, using multiple sensors increases the complexity of system setup, data collection, and sensor integration. In contrast, our *PPGResNet* model,

which relies solely on PPG data, strikes an optimal balance between high accuracy and low complexity. It demonstrates that, with the right architecture, a single signal can be leveraged effectively to achieve performance on par with (or even superior to, multi-signal approaches.

5.3 Complexity analysis

While multi-branch models typically outperform single-branch architectures in terms of accuracy, this improvement often comes with increased computational complexity and resource demands. Table 5 provides a detailed comparison of various *PPGResNet* model variants, highlighting the trade-offs between performance and efficiency. Specifically, it compares the number of parameters, memory usage, and latency for each model variant against the baseline single-branch model, *PPGResNet_3 (3).*

The baseline model, with a single convolutional branch using a kernel size of 3, contains 101,827 parameters, requires 397.76 KB of memory, and achieves an inference latency of 0.105 milliseconds. Multi-branch models, such as *PPGResNet_3* (1-3-5) and *PPGResNet_3* (3-5-7), show increased resource consumption – up to nearly four times the memory and twice the latency compared to the baseline – due to the added complexity of handling multiple receptive fields simultaneously. However, the introduction of a bottleneck architecture in the *PPGResNet_3* (1-3-5 bottleneck) variant presents a more efficient alternative. Despite using a multi-branch design, this model has only 86,275 parameters – the lowest among all the variants – and requires just 337.01 KB of memory, even less than the baseline. While its latency is slightly higher than that of the single-branch model, it remains minimal and well within practical limits.

It is also important to note that due to the 87.5% overlap used in the 8-second input windows, the system generates a new window and thus a new prediction every second. In this context, even the highest observed latency of 0.212 ms is negligible, amounting to less than 0.03% of the 1-second windowing interval. Therefore, while resource efficiency remains a design consideration, all model variants demonstrate real-time capabilities.

Table 5. Resource comparison of PPGResNet variations in terms of parameter count, memory, and latency

Model	# Parameters	Memory (KB)	Latency (ms)	Memory - relative	Latency - relative
PPGResNet_3 (3)	101,827	397.76	0.105	1.00	1.00
PPGResNet_3 (1-3-5)	253,123	988.76	0.180	2.49	1.71
PPGResNet_3 (3-5-7)	400,579	1566.72	0.212	3.94	2.01
PPGResNet_3 (3-5)	226,627	885.26	0.161	2.23	1.53
PPGResNet_3 (1-3-5 bottleneck)	86,275	337.01	0.136	0.85	1.29

6 Conclusion

This work introduced a modular, end-to-end deep-learning architecture *PPGResNet* for driver-stress detection that relies exclusively on photoplethysmography signals. The network is built from stacked multi-branch residual blocks, enabling simultaneous extraction of global and local temporal patterns without excessive depth or parameter count. An extensive experimental evaluation was conducted to determine the optimal global network architecture, serving as the backbone for establishing the optimal residual block's structure. Experimental results indicate that the proposed model achieved excellent performance, attaining accuracy exceeding 98% on two publicly available datasets. By demonstrating that a single-sensor PPG-based approach can achieve near-perfect accuracy, this work underscores the viability of lightweight, privacy-preserving stress-monitoring solutions for next-generation in-vehicle safety and comfort applications.

Although the study was restricted to PPG signals, the modular design is readily extensible to additional physiological modalities (e.g., ECG, EDA). Future research will: (i) compress and quantise the model for deployment on resource-constrained edge devices, (ii) construct a larger cross-sensor benchmark to evaluate generalisability across devices and driving conditions, and (iii) incorporate explainability techniques to enhance user trust and support real-world adoption in intelligent transportation systems.

Acknowledgement

This work has been supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia [Grant Number: 451-03-137/2025-03/ 200102].

References

[1] G. Vos, K. Trinh, Z. Sarnyai and M. Rahimi Azghadi, "Generalizable machine learning for stress monitoring from wearable devices: A systematic literature review", *International Journal of Medical Informatics*, vol. 173, pp. 105026, 2023.

[2] A. Nemcova et al., "Multimodal Features for Detection of Driver Stress and Fatigue: Review", *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3214–3233, 2021.

[3] K. Avramidis, T. Feng, D. Bose and S. Narayanan, "Multimodal Estimation Of Change Points Of Physiological Arousal During Driving", in *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, Rhodes Island, Greece, 2023, pp. 1–5.

[4] J. E. Sinex, "Pulse oximetry: Principles and limitations", *American Journal of Emergency Medicine*, vol. 17, no. 1, pp. 59–66, 1999.

[5] M. N. Rastgoo, B. Nakisa, A. Rakotonirainy, V. Chandran and D. Tjondronegoro, "A Critical Review of Proactive Detection of Driver Stress Levels Based on Multimodal Measurements", *ACM Computing Surveys*, vol. 51, no. 5, pp. 1–35, 2019.

[6] G. Xiang et al., "A multi-modal driver emotion dataset and study: Including facial expressions and synchronized physiological signals", *Engineering Applications of Artificial Intelligence*, vol. 130, pp. 107772, 2024.

[7] J. Přibil, A. Přibilová and T. Dermek, "Wearable PPG optical sensor enabling contact pressure and skin temperature measurement", *Journal of Electrical Engineering Slovak University of Technology*, vol. 76, no. 2, pp. 134–146, 2025.

[8] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks", *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[9] D. W. Otter, J. R. Medina and J. K. Kalita, "A Survey of the Usages of Deep Learning for Natural Language Processing", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, 2021.

[10] G. A. Susto, A. Cenedese and M. Terzi, "Time-Series Classification Methods: Review and Applications to Power Systems Data", in *Big Data Application in Power Systems*, Elsevier, 2018, pp. 179–220.

[11] C. Bock, M. Moor, C. R. Jutzeler and K. Borgwardt, "Machine Learning for Biomedical Time Series Classification: From Shapelets to Deep Learning", in H. Cartwright (ed.), *Artificial Neural Networks*, vol. 2190, *Methods in Molecular Biology*, Springer US, 2021, pp. 33–71.

[12] W. K. Wang et al., "A Systematic Review of Time Series Classification Techniques Used in Biomedical Applications", *Sensors*, vol. 22, no. 20, pp. 8016, 2022.

[13] P. Siirtola and J. Röning, "Comparison of Regression and Classification Models for User-Independent and Personal Stress Detection", *Sensors*, vol. 20, no. 16, pp. 4402, 2020.

[14] D. Nopez-Martinez, N. El-Haouij and R. Picard, "Detection of Real-World Driving-Induced Affective State Using Physiological Signals and Multi-View Multi-Task Machine Learning", in *2019 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, Cambridge, UK, 2019, pp. 356–361.

[15] P. Siirtola, "Continuous stress detection using the sensors of commercial smartwatch", in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, London, UK, 2019, pp. 1198–1201.

[16] M. Amin, K. Ullah, M. Asif, H. Shah, A. Mehmood and M. A. Khan, "Real-World Driver Stress Recognition and Diagnosis Based on Multimodal Deep Learning and Fuzzy EDAS Approaches", *Diagnostics*, vol. 13, no. 11, pp. 1897, 2023.

[17] V. Dissanayake et al., "SigRep: Toward Robust Wearable Emotion Recognition With Contrastive Representation Learning", *IEEE Access*, vol. 10, pp. 18105–18120, 2022.

[18] C. Szegedy et al., "Going deeper with convolutions", in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp. 1–9.

[19] V. T. Ninh et al., "An Improved Subject-Independent Stress Detection Model Applied to Consumer-grade Wearable Devices", in H. Fujita, P. Fournier-Viger, M. Ali and Y. Wang (eds.), *Advances and Trends in Artificial Intelligence*, vol. 13343, *Lecture Notes in Computer Science*, Springer International Publishing, 2022, pp. 907–919.

[20] R. Li and Z. Liu, "Stress detection using deep neural networks", *BMC Medical Informatics and Decision Making*, vol. 20, suppl. 11, pp. 285, 2020.

[21] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] A. M. Alqudah and Z. Moussavi, "A review of deep learning for biomedical signals: Current applications, advancements, future prospects, interpretation, and challenges", *Computers, Materials & Continua*, vol. 75, no. 1, pp. 1–25, 2025.

[23] L. Alzubaidi et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions", *Journal of Big Data*, vol. 8, no. 1, pp. 53, 2021.

[24] R. Pascanu, T. Mikolov and Y. Bengio, "On the difficulty of training Recurrent Neural Networks", *arXiv preprint*, 2013 Feb 15.

[25] N. M. Foumani et al., "Deep Learning for Time Series Classification and Extrinsic Regression: A Current Survey", *arXiv preprint*, 2023 Feb 5.

[26] H. Hewamalage, C. Bergmeir and K. Bandara, "Recurrent Neural Networks for Time Series Forecasting: Current Status and Future Directions", *International Journal of Forecasting*, vol. 37, no. 1, pp. 388–427, 2021.

[27] F. Ordóñez and D. Roggen, "Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition", *Sensors*, vol. 16, no. 1, pp. 115, 2016.

[28] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770–778.

[29] E. Mejía-Mejía et al., "Photoplethysmography signal processing and synthesis", in *Photoplethysmography*, Elsevier, 2022, pp. 69–146.

[30] S. Xie, R. Girshick, P. Dollar, Z. Tu and K. He, "Aggregated Residual Transformations for Deep Neural Networks", in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 5987–5995.

[31] F. Bieder, R. Sandkühler and P. C. Cattin, "Comparison of Methods Generalizing Max- and Average-Pooling", *arXiv preprint*, 2021.

[32] J. Přibil, A. Přibilová and I. Frollo, "Comparative Measurement of the PPG Signal on Different Human Body Positions by Sensors Working in Reflection and Transmission Modes", in *7th International Electronic Conference on Sensors and Applications*, 2020, p. 69.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, “Attention Is All You Need”, *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS)*, 2017 Jun 12, pp. 1–11.

[34] H. I. Fawaz et al., "InceptionTime: Finding AlexNet for Time Series Classification", *Data Mining and Knowledge Discovery*, vol. 34, no. 6, pp. 1936–1962, 2020.

[35] N. El Haouij, J.-M. Poggi, S. Sevestre-Ghalila, R. Ghozi and M. Jaïdane, "AffectiveROAD system and database to assess driver’s attention", in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, Pau, France, 2018, pp. 800–803.

[36] P. Schmidt, A. Reiss, R. Duerichen, C. Marberger and K. Van Laerhoven, "Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection", in *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, Boulder, CO, USA, 2018, pp. 400–408.

[37] C. Bustos, N. Elhaouij, A. Sole-Ribalta, J. Borge-Holthoefer, A. Lapedriza and R. Picard, "Predicting Driver Self-Reported Stress by Analyzing the Road Scene", in *2021 9th International Conference on Affective Computing and Intelligent Interaction (ACII)*, Nara, Japan, 2021, pp. 1–8.

Received 28 April 2025