IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 September 2025, accepted 27 September 2025, date of publication 6 October 2025, date of current version 10 October 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3617943

RESEARCH ARTICLE

A Lightweight 1D-CNN Architecture for Accurate and Efficient Road Type Classification Using Vibrational Signals

LORENZO MANONI, (Member, IEEE), SIMONE ORCIONI, (Senior Member, IEEE), AND MASSIMO CONTI, (Member, IEEE)

DII—Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche, 60131 Ancona, Italy

Corresponding author: Massimo Conti (m.conti@univpm.it)

This work was supported by the PNRR Italian National Center for Sustainable Mobility (MOST): SPOKE 7 "CCAM, Connected Networks and Smart Infrastructure"–WP4.

ABSTRACT This paper addresses the challenge of road type classification using deep learning techniques applied to vibrational signals collected from inertial sensors. Two novel architectures, SepRNet-1D and SepSERNet-1D, are proposed to achieve high classification accuracy while maintaining computational efficiency. The SepRNet-1D architecture is a lightweight 1D-CNN composed of multiple residual blocks, built around a Separable Convolution-1D block, that decomposes the conventional convolution operation into two distinct stages: a depthwise convolution and a pointwise convolution. SepSERNet-1D extends this design by incorporating Squeeze-and-Excitation (SE) modules to enhance feature recalibration and adaptability. Extensive experiments were conducted on a publicly available benchmark dataset, comparing the proposed architectures with state-of-the-art CNN, LSTM, hybrid CNN-LSTM models, several 2D-CNN frameworks and Transformer-based architectures. The evaluations demonstrate the superior classification performance of SepRNet-1D and SepSERNet-1D in terms of Accuracy, Precision, Recall, and F1-score. Computational experiments further highlight the lightweight design of the proposed models, achieving inference times below 4 ms in TensorFlow Lite format on a 13 GB RAM desktop CPU. The results underscore the robustness, versatility, and computational efficiency of SepRNet-1D and SepSERNet-1D, making them highly suitable for real-world road condition monitoring applications.

INDEX TERMS Deep learning, convolutional neural networks, separable convolution, road condition monitoring, vibrational signals, inertial sensors.

I. INTRODUCTION

Road condition monitoring plays a crucial role in ensuring traffic safety, preserving infrastructure, and reducing vehicle operating costs. Accurate and timely assessment of road quality is essential for road agencies to prioritize maintenance activities, optimize resource allocation, and extend the lifespan of road infrastructure. Moreover, real-time information about road conditions enhances driving comfort, reduces vehicle wear and tear, and prevents accidents caused by unexpected hazards such as potholes or uneven surfaces.

The associate editor coordinating the review of this manuscript and approving it for publication was Shan Cao.

Traditional methods of road condition monitoring often rely on visual inspections or expensive specialized equipment, which are time-consuming and lack scalability. The integration of modern sensor technologies with intelligent data processing methods has opened new opportunities for efficient and scalable road monitoring systems. Vibrational signals collected using inertial sensors, typically mounted on vehicles, provide a practical solution for detecting road anomalies and classifying road types.

Machine learning (ML) techniques, such as Support Vector Machines (SVM) [1] and Random Forests (RF) [2], have been extensively applied to process these signals. However, these methods often encounter challenges in generalizing

across diverse vehicle types, driving styles, and environmental conditions, thereby limiting their applicability in large-scale, real-world scenarios.

Recent advancements in deep learning (DL) have addressed some of these limitations, demonstrating superior performance in extracting meaningful features and achieving robust classification. This has led to the adoption of architectures such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks for road condition monitoring [3], [4]. Menegazzo et al. [5] presented an extensive and well-documented benchmark for road type classification from vibrational signals using CNN, LSTM, and hybrid CNN-LSTM networks.

Despite these advancements, there remains room for improvement in terms of generalization capabilities and classification performance, particularly when aiming to maintain computational efficiency. These aspects can be further enhanced by employing advanced and optimized DL architectures that have not yet been explored for road type classification using vibrational signals.

Furthermore, comparing the performance of DL architectures across studies remains challenging due to the lack of standardized datasets and consistency in data collection, labeling, and preprocessing methods [6], [7]. This gap underscores the need for benchmarking and standardized evaluation frameworks.

This paper addresses these shortcomings by presenting the following contributions:

- **A novel lightweight 1D-CNN architecture.** We propose a new lightweight 1D-CNN architecture, named SepRNet-1D, specifically designed for road type classification using vibrational signals.
  The proposed architecture can be regarded as a customized and optimized 1D adaptation of MobileNet V3 [8]. It combines separable convolutions and residual blocks to achieve high classification accuracy while maintaining computational efficiency.
  Furthermore, a variant of this architecture, SepSERNet-1D, integrates Squeeze-and-Excitation (SE) modules to enhance adaptability across various scenarios, including both balanced and unbalanced training and validation splits.
- **Comprehensive benchmark comparisons.** Extensive experiments were conducted on a publicly available benchmark dataset, comparing the proposed models with state-of-the-art architectures. These include CNNs, LSTMs, and hybrid CNN-LSTM models specifically designed for road type classification, as well as other widely adopted CNN frameworks–such as 2D-CNNs–and a Transformer-based architecture.
  The models' performance was evaluated under both unbalanced and balanced training/validation conditions, reflecting real-world and calibrated scenarios, respectively.
  Additionally, an extensive evaluation was performed using existing benchmark datasets with the same training and validation splits as prior studies. This approach addresses the gap in the literature regarding cross-study comparisons and promotes standardization in the field.
- **Computational experiments.** Several computational experiments were carried out to investigate the trade-off between classification performance and computational complexity of the proposed solutions. These experiments also demonstrated the potential of the proposed architectures for deployment in embedded or mobile systems, emphasizing their lightweight design and computational efficiency.

The remainder of this paper is organized as follows. Section II reviews recent literature on road condition monitoring using DL techniques and provides a coincise overview commonly employed state-of-art CNN architectures for classification tasks.

Section III introduces the proposed architectures, SepRNet-1D and SepSERNet-1D, describing their design principles based on separable convolutions and residual blocks.

Section IV describes the benchmark dataset used in this study, the preprocessing procedures applied, and the configurations of training and validation splits adopted in the experimental setup.

Section V presents and analyzes the classification results obtained across all experiments, including a comparative evaluation against models from the benchmark study as well as other state-of-the-art architectures under both unbalanced and balanced scenarios. Furthermore, the outcomes of several computational experiments are discussed to investigate the trade-offs between classification performance and computational cost of the proposed models.

Finally, Section VI concludes the paper and outlines potential directions for future research.

II. RELATED WORKS

A. LITERATURE REVIEW ON ROAD CONDITION MONITORING

The application of DL methods in road condition monitoring has achieved substantial progress due to the superior ability of neural networks to process complex data, such as vibrational signals [6]. Prior to the adoption of DL techniques, traditional ML approaches, including SVM [1], K-Nearest Neighbors (KNN) [9], Decision Trees (DT) [10], and Random Forests (RF) [2], were commonly employed for road condition assessment using vibrational data.

However, these traditional approaches often faced challenges in generalizing across diverse environmental conditions, data collection setups, acquisition platforms, and varying operational speeds. Achieving consistent performance across different contexts typically required algorithm recalibration, which constrained scalability and limited broader applicability.

In contrast, DL methods have demonstrated significant advantages over traditional ML techniques, achieving not only superior classification performance but also enhanced generalization across varied scenarios. For instance, even a simple Multi-Layer Perceptron (MLP) has been shown to outperform traditional methods such as SVM and DT. Studies by Basavaraju et al. [11] and Ferijani et al. [12] highlight the effectiveness of MLPs in classifying road abnormalities using acceleration data collected from smartphone sensors.

A significant portion of state-of-the-art research employing DL architectures, such as CNNs and LSTM networks, has focused less on the development of novel, custom deep neural network (DNN) architectures. Instead, these studies have primarily advanced in areas such as the classification of various types of road anomalies [13], the design of data collection frameworks [14], the application of advanced feature extraction techniques [15], [16], and the implementation of crowdsourced monitoring systems leveraging IoT or smartphone sensors [17].

For example, Baldini et al. [18] compared raw time-series data, Short-Time Fourier Transform (STFT), and Morlet Continuous Wavelet Transform (CWT) features as inputs to a 2D-CNN for classifying road anomalies and obstacles using acceleration and rotation data from an Xsens Mti Movella sensor. Similarly, Varona et al. [3] utilized 3-axis smartphone accelerometer data with CNN, LSTM, and Reservoir Computing models to distinguish road anomalies from acceleration changes caused by driver actions.

In contrast, some studies have proposed hybrid architectures and data fusion methods to enhance DNN performance. For instance, Raslan et al. [19] introduced a lightweight hybrid model, SepConv1D-BiLSTM, for classifying road segments (e.g., normal, speed bump, pothole, or bad road) using time-series data, including 3-axis acceleration, rotation, and speed. The Separable-1D-Conv component processed raw signals, while the BiLSTM component utilized features extracted via FFT transformation.

Similarly, Singh et al. [4] incorporated a binary feature extractor within an LSTM network to detect potholes using smartphone accelerometer data.

Further innovations include the hybrid architectures proposed by Pandey et al. [20], which combine image and acceleration data using a hybrid 2D-CNN + 1D-CNN network for pothole detection. Data collection was performed using a smartphone mounted on the windshield, capturing 3-axis acceleration at 100 Hz and images simultaneously. Additionally, Xin et al. [21] developed a novel crowdsourcing framework for pothole detection, integrating 3-axis smartphone accelerometer data with video frame features extracted via multiple convolutional layers.

Conversely, Doz et al. [22] compared the performance of several time-frequency representations for road type classification using signals collected with piezoelectric sensors, employing both ML and DL techniques.

One of the few publicly available datasets in this domain was introduced by Menegazzo et al. [5], providing an extensive and well-documented benchmark for road type classification from vibrational signals collected with inertial sensors. The benchmark includes dataset publication, code for training and testing custom architectures, and models such as CNN, LSTM, and hybrid CNN-LSTM for classifying road types (e.g., asphalt, cobblestone, and dirt). This benchmark represents a critical step toward standardizing performance comparisons and has been selected for the experiments in this paper.

However, a prior comparison of state-of-the-art models by Narit et al. [23] revealed several shortcomings in ensuring fairness and consistency with the original architectures described in Menegazzo et al. [5]. For instance, it remains unclear whether the same training/validation split was used, and the proposed architecture lacks key details, including the number of channels, filter sizes, dropout rates, and convolution types (1D or 2D).

B. STATE-OF-THE-ART CNN ARCHITECTURES

This section provides a concise overview of some of the most prominent and widely adopted DL architectures, that have established benchmarks in the field of classification, offering insights into their design principles and capabilities. The analysis focuses on several DL frameworks, including some commonly used CNN architectures and Transformer-based models. Although these models were not originally designed for road condition monitoring, they incorporate key design principles, such as residual connections, separable convolutions, and self-attention mechanisms, that make them suitable for time-series classification tasks.

Certain architectures were specifically designed for real-time implementation on mobile or embedded platforms, prioritizing lightweight deployment. Conversely, others emphasize flexibility and scalability for computationally intensive tasks.

- **LeNet** [24]. LeNet, one of the earliest CNN architectures, is renowned for its simplicity and efficiency. Its structure consists of convolutional layers, pooling layers, and fully connected layers, making it well-suited for tasks requiring moderate computational resources. Although relatively basic compared to modern architectures, LeNet remains a benchmark in the field due to its low computational overhead and suitability for real-time implementation. For instance, it has been used for comparison in [25].
- **ResNet** [26]. ResNet revolutionized DL architectures with the introduction of residual blocks featuring skip connections. These blocks consist of a main path, where multiple convolutional operations are applied to the input features, and a skip path, which adjusts the number of channels to enable summation with the main path's output. ResNet was developed in various depths, including 18, 34, 50, 101, and 152 layers, offering flexibility based on the computational requirements of

the task. Its scalable design has made it a cornerstone for tasks of varying complexity.

- **Xception** [27]. Xception builds on the Inception architecture [28] by incorporating separable convolutions into residual blocks to significantly reduce computational cost. Separable convolutions divide the standard convolution operation into two steps: filtering and combining, resulting in a more parameter-efficient model without sacrificing accuracy. This architecture is particularly well-suited for high-performance tasks with constrained computational resources.
- **ShuffleNet V2** [29]. ShuffleNet V2 is a lightweight architecture optimized for mobile and embedded systems, emphasizing efficient computation and minimal memory usage. It achieves this through residual blocks with separable convolutions and innovative techniques such as channel splitting and channel shuffling, which ensure balanced feature processing while minimizing computational overhead. ShuffleNet V2 is ideal for applications requiring high accuracy in resource-constrained environments.
- **Transformer** [30]. Initially developed for sequence modeling in natural language processing, Transformer models have recently gained traction in time-series classification tasks. Their core mechanism, self-attention, allows the model to evaluate relationships between all time steps in a sequence simultaneously. This allows for the extraction of long-range dependencies without relying on recurrence, making Transformer encoders particularly effective for time-series application. The architecture proposed in [30] comprises a stack of four 1D convolutional layers followed by six Transformer encoder blocks. These blocks generate the feature representation used by the final dense layer to produce output probabilities.
  This architecture was chosen for comparison in a recent study conducted by Raslan et al. [19], which introduced an hybrid SepConv1D-BiLSTM model for road anomaly classification using vibrational signals. Among the state-of-the-art models evaluated in that work, the Transformer model achieved the best classification performances, demonstrating its effectiveness and suitability for this task.
- **MobileNet V3** [8]. MobileNet V3 represents a significant advancement in lightweight CNNs, leveraging separable convolutions alongside the inverted residual bottleneck structure and Squeeze-and-Excitation modules for improved efficiency and performance. The inverted residual bottleneck expands input channels through pointwise convolutions, processes the expanded features using depthwise separable convolutions, and subsequently reduces the feature dimensions. This design delivers a high-accuracy architecture optimized for computational efficiency, making it well-suited for mobile and embedded platforms.

III. PROPOSED ARCHITECTURE

The proposed 1D-CNN lightweight architecture was developed through a principled adaptation of established DL components, such as separable convolutions, residual connections and SE modules, to the specific characteristics of 1D vibrational signals. This process involved non-trivial architectural decisions, including the design of residual connections and separable convolution blocks, as well as the selection and positions of SE modules. These choices were guided by empirical effectiveness and their compatibility with the temporal nature of inertial data.

A. SEPARABLE 1D CONVOLUTIONS

The proposed architecture is structured around residual blocks, that incorporate separable one-dimensional convolutions. Separable convolutions decompose the conventional convolution operation into two distinct stages: a depthwise convolution, which performs spatial filtering across individual input channels, and a pointwise convolution, which modifies the number of output channels by combining the results across feature maps.

In this architecture, reported in Fig. 1, the one-dimensional separable convolution block (Separable Conv1D) takes as input a feature map $X \in \mathbb{R}^{W \times C_{\text{in}}}$. The process begin with a depthwise convolution layer (Depthwise Conv1D) employing C_{in} filters of size D, with corresponding weights stored in the tensor $K^{\text{depth}} \in \mathbb{R}^{D \times C_{\text{in}}}$. This operation is followed by a Rectified Linear Unit activation (ReLU) and a Batch Normalization layer (BatchNorm). Next a pointwise convolution (Pointwise Conv1D), is applied to project the feature representation from C_{in} to C_{out} channel using the weights matrix $K^{\text{point}} \in \mathbb{R}^{C_{\text{in}} \times C_{\text{out}}}$. As with the previous stage, this operation is followed by ReLU and BatchNorm layers.

These operations can be mathematically expressed as,

$$Z_{t,c} = \mathcal{B}_{\mathcal{N}}\left(\mathcal{R}\left(\sum_{u=1}^{D} X_{t-u,c} K_{u,c}^{\text{depth}} + b_c\right)\right) \tag{1}$$

$$Y_{t,s} = \mathcal{B}_{\mathcal{N}}\left(\mathcal{R}\left(\sum_{c=1}^{C_{\text{in}}} Z_{t,c} K_{c,s}^{\text{point}} + b_s\right)\right) \times \forall t = 1, \ldots, W;\ \forall c = 1, \ldots, C_{\text{in}};\ \forall s = 1, \ldots, C_{\text{out}}; \tag{2}$$

where $\mathcal{R}(\cdot)$ represents the ReLU activation function and $\mathcal{B}_{\mathcal{N}}(\cdot)$ denotes the Batch Normalization operator. The terms of X corresponding to negative matrix indices in depthwise convolution are handled through zero-padding.

This structure significantly reduces the computational cost compared to standard convolution by approximately a factor proportional to the filter size D.

The number of sums and multiplications of Eqs. 1, 2, and of the standard convolution are DC_{in}, $C_{\text{in}}C_{\text{out}}$ and $DC_{\text{in}}C_{\text{out}}$, respectively. Therefore, the reduction in computational

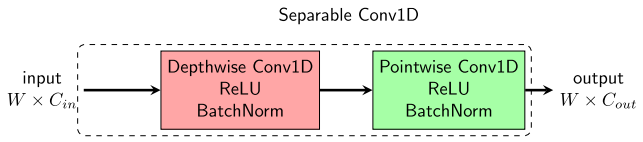


FIGURE 1. Structure of the 1D separable convolution block for the proposed architecture.

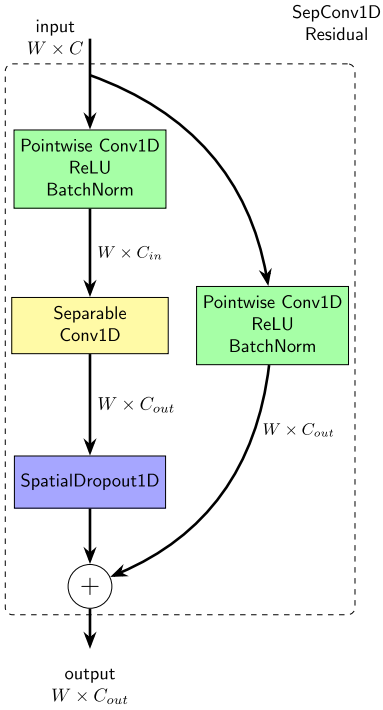


FIGURE 2. Structure of the 1D residual block based on separable convolutions in the proposed SepRNet-1D architecture.

complexity can be expressed as:

$$\text{FLOP}_{\text{ratio}} = \frac{D{\cdot}C_{\text{in}} + C_{\text{in}}C_{\text{out}}}{D{\cdot}C_{\text{in}}C_{\text{out}}} = \frac{D + C_{\text{out}}}{D{\cdot}C_{\text{out}}} \approx \frac{1}{D} \quad (3)$$

B. SepRNet-1D ARCHITECTURE

The proposed SepRNet-1D architecture is a lightweight 1D-CNN composed of multiple residual blocks, built around the previously described Separable Conv1D block.

The structure of these blocks, referred to as SepConv1D Residual, is presented in Fig. 2. Each block processes an input feature $X \in \mathbb{R}^{W\times C}$. A pointwise convolution, followed by ReLU and BatchNorm layers, initially adjusts the number of channels from C to C_{in}. Thereafter, a Separable Conv1D block further refines the feature map, increasing the number of channels to C_{out} channels. To improve generalization and mitigate overfitting, a SpatialDropout1D layer is applied to the output feature $Y \in \mathbb{R}^{W\times C_{\text{out}}}$. The incorporation of a SpatialDropout operator was a pivotal design choice, as it provides effective regularization by randomly dropping entire spatial feature maps during training. This approach contrasts with traditional Dropout, which operates at the individual feature element level. SpatialDropout has been shown to outperform traditional Dropout in scenarios with high inter-channel correlations [31], [32].

The proposed SepRNet-1D, shown in Fig. 3, can be viewed as an extremely lightweight 1D adaptation of MobileNet V3. It processes an input multi-dimensional time series $X_{in} \in \mathbb{R}^{W\times N_{var}}$, where N_{var} represents the number of variables (7 in this case), through an initial convolution comprising a Conv1D, ReLU, BatchNorm and SpatialDropout1D layers.

This first block is then followed by n residual blocks (SepConv1D Residual), each employing separable convolutions with varying numbers of internal channels C_{in}, output channels C_{out}, kernel size D, and spatial dropout rate r. A Global Averaged Pooling (GAP) layer then computes the mean across the spatial axis, producing an output vector of size $C_{\text{out}}^{(n)}$.

The GAP layer was selected over a Flatten layer, as it significantly reduces trainable parameters and overall computational complexity. Additionally, it serves as an effective strategy to mitigate overfitting by limiting the model's capacity and promoting generalization [33].

The output vector of the GAP layer is passed through a Dropout layer, followed by a fully connected Dense layer with U units, a ReLU, and BatchNorm layers. Finally, a second Dense layer followed by a Softmax activation layer produces the predicted class probabilities for the input signal.

Similarly to MobileNet V3, the SepRNet-1D architecture employs stacked residual units with depthwise separable convolutions. However, the proposed network features customized lightweight 1D separable convolution blocks with larger kernel size D and a reduces number of filters.

In contrast to the residual bottleneck of MobileNet V3, the number of internal channels C_{in} in SepRNet-1D is not constrained to be an expansion of the input channels ($\alpha C, \alpha > 1$). Additionally, a maximum of $n = 3$ residual blocks was used, ensuring an extremely compact network design.

C. SepSERNet-1D: IMPROVED SepRNet-1D WITH SQUEEZE AND EXCITATION MODULE

In addition to the SepRNet-1D architecture described earlier, this work introduces an alternative version called SepSERNet-1D, which modifies the structure of the residual blocks by incorporating a SE module [34].

Given a feature input $X \in \mathbb{R}^{W\times C_{\text{in}}}$, the SE block reduces its spatial dimension to 1 applying a GAP across the spatial dimension. This operation produces a vector $z \in \mathbb{R}^{C_{\text{in}}}$, which encapsulates the channel-wise global context of the input features.

$$z_c = \frac{1}{W}\sum_{t=1}^{W} X_{t,c} \quad (4)$$

Thereafter, a reduced size-vector $f \in \mathbb{R}^{C_{\text{in}}/r}$ is obtained by passing z through a fully connected (Dense) layer, where

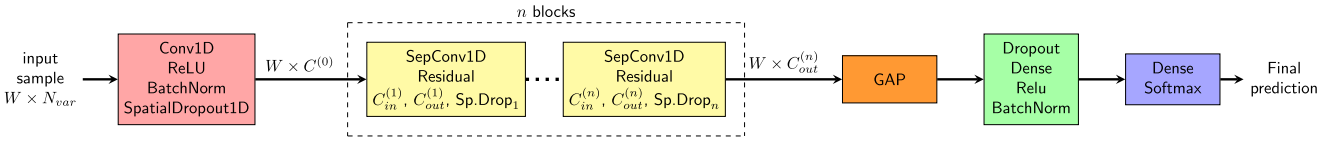


FIGURE 3. General architecture of the proposed SepRNet-1D.

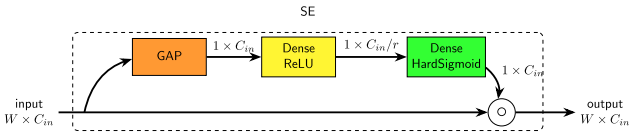


FIGURE 4. Structure of 1D SE block used in SepSERNet-1D.

$r > 1$ acts as the reduction ratio, followed by a ReLU layer. The size of the vector is then restored by applying to f another Dense layer, followed by HardSigmoid to produce an output vector $g \in \mathbb{R}^{C_{\text{in}}}$. Finally, the output feature map is computed by performing a channel-wise multiplication between the input feature map X and the vector g, effectively reweighting the input channels based on their importance.

The transformations to compute the output feature map $Y \in \mathbb{R}^{W \times C_{\text{in}}}$, which has the same size as the input, can be expressed mathematically as,

$$\begin{aligned} f &= \mathcal{R}\left(W^T z_c + b_W\right) \\ g &= \sigma_h\left(V^T f + b_V\right) \\ Y &= X \circ g \end{aligned} \tag{5}$$

where $W \in \mathbb{R}^{C_{\text{in}} \times C_{\text{in}}/r}$, $V \in \mathbb{R}^{C_{\text{in}}/r \times C_{\text{in}}}$ are the weight matrices of to the two FC layers, $\circ$ is the Hadamard product and $\sigma_h(\cdot)$ is the hard sigmoid function

$$\sigma_h(x) = \max\left(0, \min\left(1, \frac{x+3}{6}\right)\right) \tag{6}$$

Figure 4 illustrates the described SE block, summarizing the transformations and their role in recalibrating channel-wise features.

The hard sigmoid activation was selected for its computational efficiency and suitability for embedded applications, as demonstrated in MobileNetV3 [8], where it contributes to reduced inference time without significantly compromising performance.

SepSERNet-1D incorporates SE modules into the residual blocks immediately after the depthwise convolution. This placement was carefully chosen because SE modules primarily function to reweight feature maps, emphasizing the most relevant ones while suppressing less critical features. If SE modules were placed after the pointwise convolution, the effect of feature reweighting would be compromised by the subsequent application of SpatialDropout1D, which randomly removes features during training.

The structure of the modified residual block incorporating the SE module is illustrated in Fig. 5.

TABLE 1. Description of the nine PVS datasets.

Dataset	Vehicle	Driver	Scenario
PVS 1	Volkswagen Saveiro	Driver 1	Scenario 1
PVS 2	Volkswagen Saveiro	Driver 1	Scenario 2
PVS 3	Volkswagen Saveiro	Driver 1	Scenario 3
PVS 4	Fiat Bravo	Driver 2	Scenario 1
PVS 5	Fiat Bravo	Driver 2	Scenario 2
PVS 6	Fiat Bravo	Driver 2	Scenario 3
PVS 7	Fiat Palio	Driver 3	Scenario 1
PVS 8	Fiat Palio	Driver 3	Scenario 2
PVS 9	Fiat Palio	Driver 3	Scenario 3

TABLE 2. Consistency across classes for the splits of the three experiments.

	Asphalt road		Cobblestone road		Dirt road	
	Train	Val	Train	Val	Train	Val
Exp. 1	1948	1164	1708	384	996	998
Exp. 2	2134	978	1374	718	1276	718
Exp. 3	2066	1046	1442	650	1238	756

IV. DATASET DESCRIPTION

The dataset used in this study was originally collected by [5] for the classification of road types using vibrational signals recorded with inertial sensors. The data collection setup included multiple MPU-9250 units strategically positioned on both the left and the right sides of the vehicle, a Raspberry Pi 3 for signal storage, a camera for video acquisition and a Global Positioning System (GPS) module.

The inertial units were placed in three different positions: on the dashboard, above the suspension, and below suspension, on both the left and right sides of the vehicle. Each MPU-9250 unit was equipped with an accelerometer, gyroscope, magnetometer, and temperature sensor. The sampling frequency for the MPU sensors (accelerometer, gyroscope, temperature, and magnetometer) was set to 100 Hz, while the GPS had a sampling rate of 1 Hz. The camera recorded at a frame rate of 30 Hz.

The response in terms of acceleration and rotation rate resulting from road irregularities depends not only on the placement of the Inertial Measurement Unit (IMU) sensors but also on the vehicle properties and driving style.

Data collection was conducted by driving along three distinct sets of routes, referred to as 'scenarios'. To enhance the generalization capabilities of the classification algorithm, nine different datasets, named Passive Vehicular Sensors Dataset (PVS 1-9) in [5], were collected across the three

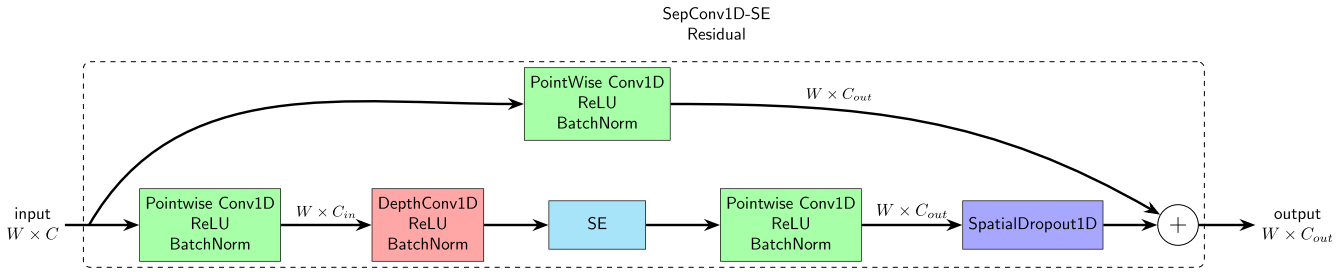


FIGURE 5. Structure of the modified residual block with insertion of SE module.

TABLE 3. Tested configurations of the proposed SepRNet-1D ($W \times 300$).

Config	Number of blocks	Init filters	Init kernel	Init dropout	Input channels C	Internal channels C_{in}	Output channels C_{out}	Kernel sizes	Dropout rates	Final dropout rate	Final units
Config #1	2	64	7	0.1	64,64	64,64	64,64	7,7	0.75,0.75	-	-
Config #2	2	64	7	0.1	64,128	64,128	128,128	7,7	0.6,0.6	-	-
Config #3	3	64	7	0.1	64,64,64	64,64,64	64,64,64	7,7,7	0.85,0.85,0.85	-	-
Config #4	2	64	7	0.1	64,256	64,256	256,256	7,7	0.3,0.3	-	-
Config #5	2	64	7	0.1	64,160	64,160	160,160	7,7	0.25,0.25	-	-
Config #6	1	64	7	0.1	64	64	256	7	0.6	-	-
Config #7	2	64	7	0.1	64,128	64,128	128,128	7,7	0.6,0.6	0.4	64
Config #8	2	64	7	0.1	64,128	64,128	128,128	3,3	0.6,0.6	-	-
Config #9	2	64	7	0.1	64,128	256,512	128,128	7,7	0.6,0.6	-	-
Config #10	2	64	7	-	64,128	64,128	128,128	7,7	-,-	-	-

TABLE 4. Results in terms of Accuracy, Precision, Recall and F1-score of the proposed SepRNet-1D configurations for all experiments.

	Exp.	Cfg #1	Cfg #2	Cfg #3	Cfg #4	Cfg #5	Cfg #6	Cfg #7	Cfg #8	Cfg #9	Cfg #10
Accuracy	1	96.15	96.78	96.07	96.66	97.00	96.31	95.60	95.84	**97.31**	95.44
	2	91.88	92.83	91.09	91.71	91.84	92.13	91.22	**92.92**	92.17	91.51
	3	93.80	**94.45**	94.00	94.17	94.37	93.72	94.45	93.11	94.13	93.47
	Avg	93.94	**94.69**	93.72	94.18	94.40	94.05	93.76	93.96	94.54	93.47
Precision	1	94.97	95.32	94.00	94.91	95.67	95.12	93.47	93.51	**95.83**	93.28
	2	91.22	92.10	91.05	91.04	91.16	91.46	90.29	**92.34**	91.71	90.96
	3	92.87	93.68	93.39	93.56	93.54	92.90	**93.81**	92.16	93.41	92.87
	Avg	93.02	**93.70**	92.97	93.17	93.46	93.16	92.52	92.67	93.65	92.37
Recall	1	94.10	95.47	94.00	95.87	95.88	94.00	93.89	94.25	**96.73**	93.32
	2	90.96	91.12	90.09	90.91	90.08	91.24	90.19	**92.11**	91.30	90.58
	3	92.84	**93.66**	92.88	93.56	93.50	92.84	93.51	92.19	93.17	92.18
	Avg	92.63	**93.75**	92.32	93.45	93.15	92.69	92.53	92.85	93.73	92.03
F1-score	1	94.51	95.39	94.22	95.37	95.78	94.52	93.68	93.86	**96.26**	93.30
	2	91.00	92.11	90.03	90.85	91.00	91.24	90.24	**92.16**	91.29	90.53
	3	92.89	93.66	93.03	93.21	93.52	92.84	**94.00**	92.09	93.29	92.39
	Avg	92.80	**93.72**	92.44	93.14	93.43	92.69	92.52	92.70	93.61	92.07

scenarios, featuring various combinations of car models and drivers, as detailed in Tab. 1.

Data collected from each individual MPU unit, placed at specific locations and sides (left or right) within the vehicle, included 3-axis acceleration (A_x, A_y, A_z), 3-axis rotation rate (G_x, G_y, G_z), vehicle speed, temperature, and GPS coordinates.

Labeling was performed by classifying individual raw-time domain signal samples into one of three road types: asphalt road, cobblestone road, or dirt road. Fig 6 presents sample frames of acceleration and rotation rate signals recorded by the sensors placed on the dashboard. Additionally, in the study conducted by [5] a categorization for speed bump detection was performed, which was later used in [35].

A. PREPROCESSING AND SPLITTING

To ensure a fair comparison with the results obtained using the DL techniques proposed in [5], the same preprocessing

TABLE 5. Results in terms of Accuracy, Precision, Recall and F1-score of CNN models proposed in [5].

	Exp.	CNN 1	CNN 2	CNN 3	CNN 4	CNN 5	CNN 6	CNN 7	CNN 8
Accuracy	1	89.17	90.3	95.33	93.48	89.28	**95.50**	92.50	93.56
	2	85.17	84.63	89.64	87.49	84.59	87.40	87.70	**90.64**
	3	87.15	87.23	**93.47**	92.58	89.64	93.10	92.66	93.15
	Avg	87.40	87.39	**92.81**	91.18	87.84	92.00	90.95	92.45
Precision	1	86.01	86.35	92.84	90.79	84.48	**94.14**	88.57	91.00
	2	84.44	83.57	88.87	87.39	84.30	86.30	86.64	**89.68**
	3	85.26	85.43	**92.71**	91.98	88.26	92.18	91.78	92.22
	Avg	85.24	85.12	**91.47**	90.05	85.68	90.87	89.00	90.86
Recall	1	86.58	87.37	**93.72**	90.62	84.39	92.67	90.23	91.06
	2	83.56	82.89	88.44	86.07	82.90	86.07	86.26	**90.00**
	3	85.42	85.47	**92.28**	91.10	88.14	91.96	91.40	90.23
	Avg	85.22	85.24	**91.48**	89.26	85.14	90.23	89.30	90.31
F1-score	1	86.20	86.79	93.26	90.64	84.39	**93.38**	89.28	90.83
	2	83.44	82.71	88.43	85.86	82.50	86.03	86.33	**89.65**
	3	85.29	85.30	**92.45**	91.31	88.10	92.07	91.55	92.12
	Avg	84.69	84.93	**91.38**	89.27	85.00	90.49	89.05	90.87

TABLE 6. Results in terms of Accuracy, Precision, Recall and F1-score of LSTM models proposed in [5].

	Exp.	LSTM 1	LSTM 2	LSTM 3	LSTM 4	LSTM 5	LSTM 6	LSTM 7
Accuracy	1	70.15	83.03	82.80	87.47	74.78	81.87	**92.62**
	2	85.55	85.29	85.67	85.75	80.45	88.94	**91.80**
	3	90.33	87.64	90.13	87.40	81.73	90.38	**92.09**
	Avg	82.01	85.32	86.20	86.87	78.99	87.10	**92.17**
Precision	1	67.12	78.35	77.68	82.20	70.18	76.98	**89.24**
	2	84.26	84.07	84.56	84.68	78.84	87.86	**91.05**
	3	88.87	85.83	88.56	85.89	80.05	89.32	**91.06**
	Avg	80.08	82.75	83.60	84.26	76.36	84.72	**90.45**
Recall	1	63.18	65.46	80.38	84.87	70.81	78.75	**90.21**
	2	83.88	83.78	84.10	84.39	78.86	87.83	**90.94**
	3	88.90	85.99	88.71	85.86	80.08	89.27	**90.67**
	Avg	80.32	78.41	84.40	85.04	76.58	85.28	**90.61**
F1-score	1	65.97	65.87	78.30	83.01	69.51	77.50	**89.69**
	2	83.97	83.70	83.98	84.25	78.83	87.80	**90.93**
	3	88.88	85.90	88.64	85.62	79.86	89.03	**90.79**
	Avg	79.61	76.72	83.64	84.29	76.07	84.78	**90.43**

technique and training/validation splits as those used in [5] were adopted.

First, a component-wise normalization was applied to the individual total signals collected for each of the nine PVS datasets. In [5] three different normalizers were considered:

- Min-Max Normalizer which scales the values to the range (0, 1);
- Min-Max Normalizer which scales the values to the range $(-1, 1)$;
- Robust Normalizer which removes the mean and scales the data based on the interquartile range (IQR).

Since the Min-Max Normalizer yielded the best classification performance in [5], the same normalizer was chosen in this paper.

Similarly to the preprocessing steps in [5], the signals from each PVS subset were segmented into non-overlapping frames. Each frame which were labeled the most common label value in the corresponding window.

A fixed window length of 300 samples, corresponding to a 3 s time window, was used in this study as it provided the highest classification metrics for the best DL architectures of each group: CNN, LSTM and CNN-LSTM.

Following the approach in [5], in this work the 3 acceleration components (A_x, A_y, A_z), the 3 components of rotational velocity (G_x, G_y, G_z) and also vehicle speed were selected to construct the training and validation sets. Consequently, the segmentation of the multidimensional time-series yields frames of size $W \times N_{var}$, where $W = 300$ and $N_{\text{var}} = 7$.

TABLE 7. Results in terms of Accuracy, Precision, Recall and F1-score of CNN-LSTM models proposed in [5].

	Exp.	CNN-LSTM 1	CNN-LSTM 2	CNN-LSTM 3	CNN-LSTM 4	CNN-LSTM 5	CNN-LSTM 6
Accuracy	1	91.91	91.48	90.07	91.56	**93.95**	93.24
	2	87.74	87.57	87.66	85.29	90.39	**90.64**
	3	91.39	91.39	91.15	90.99	93.52	**94.09**
	Avg	90.35	90.15	89.93	89.28	92.62	**92.66**
Precision	1	87.68	87.42	87.06	87.36	**90.97**	89.61
	2	86.69	86.81	86.94	84.42	89.64	**90.40**
	3	90.11	90.16	89.87	89.70	92.84	**93.36**
	Avg	88.16	88.13	87.96	87.16	**91.15**	91.12
Recall	1	90.21	88.72	87.78	87.86	91.17	**92.32**
	2	86.33	86.17	86.29	83.64	89.32	**89.59**
	3	90.06	90.31	89.68	89.51	92.35	**93.13**
	Avg	88.87	88.40	87.92	87.00	90.95	**91.68**
F1-score	1	88.63	88.00	87.39	87.60	**91.07**	90.71
	2	86.29	86.08	86.20	83.43	89.30	**89.51**
	3	90.07	90.14	89.75	89.60	92.51	**93.22**
	Avg	88.33	88.07	87.78	86.88	90.96	**91.15**

TABLE 8. Classification results of the best configuration of SepRNet-1D (Cfg #2 in Tab. 4) and SepSERNet-1D, which shares the same configuration parameters.

	Accuracy				Precision				Recall				F1-score			
Experiment	**1**	**2**	**3**	**Avg.**	**1**	**2**	**3**	**Avg.**	**1**	**2**	**3**	**Avg.**	**1**	**2**	**3**	**Avg.**
SepRNet-1D	96.78	92.83	94.45	94.69	95.32	92.1	93.68	93.70	95.47	92.12	93.66	93.75	95.39	92.11	93.66	93.72
SepSERNet-1D	96.23	91.34	94.25	93.94	94.95	90.46	93.54	92.98	93.99	90.36	93.68	92.68	94.44	90.40	93.61	92.82

TABLE 9. Results in terms of classification metrics, averaged across experiments for each category, corresponding to the best configuration within each model group.

	Asphalt road			Cobblestone road			Dirt road		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
SepRNet-1D	98.84	98.87	98.85	**88.88**	**89.84**	**89.36**	**93.60**	**92.84**	**93.22**
SepSERNet-1D	98.68	98.37	98.52	88.66	86.53	87.58	91.68	93.61	92.63
Best CNN [5]	98.60	**99.15**	**98.87**	88.17	82.08	85.01	88.70	92.39	90.51
Best LSTM [5]	98.84	98.65	98.74	85.88	82.30	84.06	88.00	90.82	89.39
Best CNN-LSTM [5]	**98.86**	98.31	98.58	87.03	82.93	84.93	88.69	92.31	90.46

Since vehicle speed strongly influences vehicle acceleration response to road irregularities, the inclusion of the vehicle speed in input data represents an appropriate choice to enhance the classification performance of the DL model.

In this paper, signals collected from the dashboard on both sides of the vehicle were selected for experiments. In the context of a real-world application, sensor positioning on the dashboard (rather than near suspension) offers a much more practical and commonly employed solution in road condition monitoring systems. This is particularly relevant for crowdsourced monitoring systems, where sensor installation near suspension might be intrusive.

The same three training/validation splits (denoted as Experiment 1, 2, 3) as in [5] were applied, with a subset of the nine PVS datasets assigned for training and the remaining data used for validation.

- **Exp. 1** Training: PVS 1, 3, 4, 6, 7, 9; Test: PVS 2, 5, 8.
- **Exp. 2** Training: PVS 1, 2, 3, 7, 8, 9; Test: PVS 4, 5, 6.
- **Exp. 3** Training: PVS 1, 2, 4, 6, 8, 9; Test: PVS 3, 5, 7.

A reasonable criterion was applied for assigning the PVS subsets to either training or validation sets. In Experiment 1, the training data included signals from all vehicles (with their corresponding drivers) for scenarios 1 and 3, while the validation data consisted of signals from the same vehicles but collected in a different subsets of roads, namely scenario 2.

In contrast, for Experiment 2, data from all scenarios were included in both the training and validation set. However, the

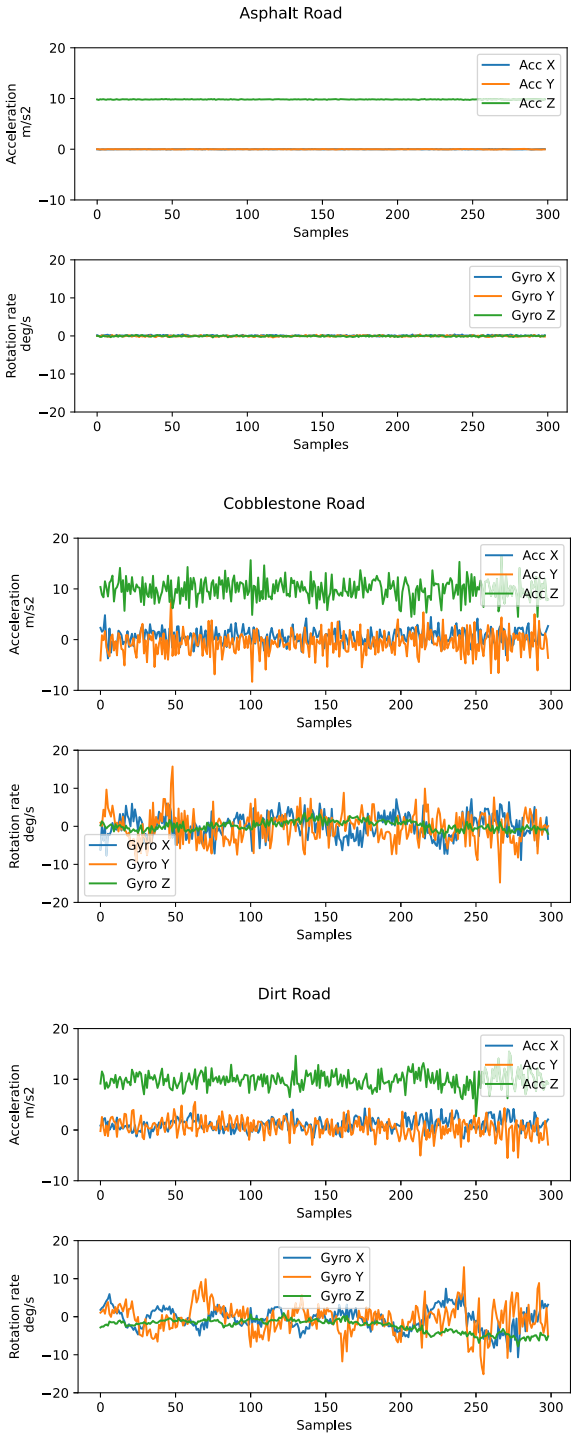
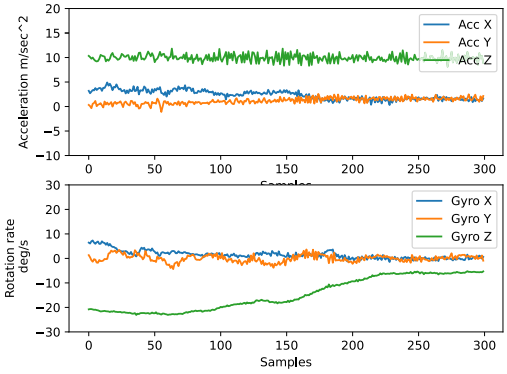


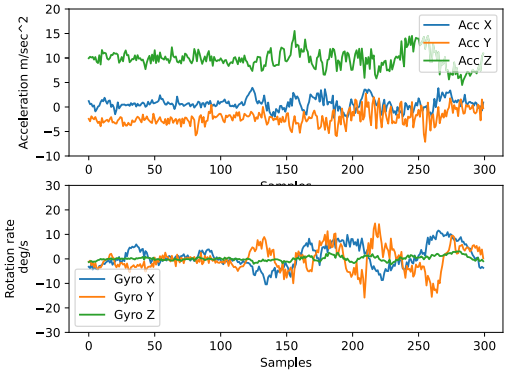
FIGURE 6. Frames of collected signals with sensors placed on left side of dashboard by driving on each of the three road types on PVS-3.

validation data were collected using a different vehicle than the one used for training.

For Experiment 3, a more heterogeneous split was chosen. Both the training and validation data included signals of all vehicles, but for each vehicle the training and validation sets were collected in different scenarios.



(a) classified as cobblestone road



(b) classified as dirt road

FIGURE 7. Acceleration and angular velocity of two examples of misclassified frames collected by driving on asphalt road.

Tab. 2 presents the number of segmented frames with a length of 300 samples, belonging to the three classes for the three experiments.

V. EXPERIMENTAL RESULTS

To demonstrate the effectiveness and the superiority of the proposed model over the state-of-the-art existing architectures, an extensive comparison was conducted with all the configurations of CNN, LSTM and CNN-LSTM models used in [5]. Additionally, the classification performance was compared to that of by several well known and widely used state-of-the-art CNN architectures, all of which were trained and tested on the same training/validation splits as SepRNet-1D.

All models were implemented in TensorFlow with a Keras backend (version 2.15.0), in a Python 3.10.12 environment on a Google Colab Platform, including a NVIDIA Tesla T4 GPU. For all configurations of the proposed model and across the three experiments, the Adam optimizer was employed with a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a batch size of 64. Training was terminated after 50 epochs without any improvement in validation accuracy.

Furthermore, an additional experiment was conducted by training and testing all models on a custom-generated,

TABLE 10. Results in terms of classification metrics, averaged across all experiments, obtained by SepRNet-1D, SepSERNet-1D, the best configurations of CNN, LSTM, CNN-LSTM, and by other state-of-the-art architectures.

Model	Accuracy	Precision	Recall	F1-score
SepRNet-1D	**94.69**	**93.70**	**93.75**	**93.72**
SepSERNet-1D	93.94	92.98	92.68	92.82
Best CNN [5]	92.81	91.47	91.48	91.38
Best LSTM [5]	92.17	90.45	90.61	90.43
Best CNN-LSTM [5]	92.66	91.12	91.68	91.15
LeNet-5 [24]	88.38	85.98	86.53	86.16
Xception [27]	92.49	91.35	90.75	91.17
Truncated ResNet18 [26]	91.30	89.91	89.48	89.54
ShuffleNet v2 0.25x [29]	88.53	86.52	84.97	85.45
Tiny ShuffleNet v2 0.25x [29]	88.47	86.21	86.75	86.07
Transformer [30]	92.40	91.14	91.16	91.04
Transformer tiny [30]	91.14	89.20	89.16	89.67
MobileNet V3 small [8]	90.02	88.56	87.60	87.89
Trunc. MobileNet V3 small [8]	93.12	91.17	92.12	91.80

TABLE 11. Results in terms of classification metrics, averaged across all experiments for each category obtained by SepRNet-1D,SepSERNet-1D, the best configurations of CNN, LSTM, CNN-LSTM and by the other state-of-the-art architectures.

	Asphalt road			Cobblestone			Dirt road		
Model	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
SepRNet-1D	98.84	98.87	98.85	**88.88**	**89.84**	**89.36**	**93.60**	**92.84**	**93.22**
SepSERNet-1D	98.68	98.37	98.52	88.66	86.53	87.58	91.68	93.61	92.63
Best CNN [5]	98.60	**99.15**	**98.87**	88.17	82.08	85.01	88.70	92.39	90.51
Best LSTM [5]	98.84	98.65	98.74	85.88	82.30	84.06	88.00	90.82	89.39
Best CNN-LSTM [5]	**98.86**	98.31	98.58	87.03	82.93	84.93	88.69	92.31	90.46
LeNet-5 [24]	98.40	96.80	97.60	76.86	77.91	77.38	84.16	85.11	84.63
Xception [27]	98.00	98.56	98.28	85.96	84.24	85.10	90.00	90.61	90.30
Truncated ResNet18 [26]	98.37	98.49	98.43	86.77	77.51	81.81	85.61	91.95	88.67
ShuffleNet v2 0.25x [29]	97.90	98.06	97.98	79.39	73.46	76.31	82.76	86.97	84.81
Tiny ShuffleNet v2 0.25x [29]	98.58	98.02	98.30	77.45	75.45	76.39	83.42	85.48	84.48
Transformer [30]	98.61	98.49	98.53	87.16	82.53	84.78	88.20	91.67	89.90
Transformer tiny [30]	98.37	98.49	98.43	83.37	81.28	82.31	87.31	88.71	88.00
MobileNet V3-small [8]	98.35	97.49	97.92	82.04	78.20	80.07	85.05	88.83	86.90
Trunc. MobileNet V3-small [8]	98.84	98.90	**98.87**	86.58	85.54	86.01	90.45	91.22	90.83

TABLE 12. Consistency of training and validation splits of Custom Experiment.

	Asphalt road	Cobblestone road	Dirt road	All classes
Training	2198	1466	1374	5038
Validation	914	626	620	2160
Total	3112	2092	1994	7198

TABLE 13. Number of sample for each PVS subsets for training and validation slits of Custom Experiment.

	Training	Validation	Total
PVS-1	677	283	960
PVS-2	572	258	836
PVS-3	478	226	704
PVS-4	627	255	882
PVS-5	643	249	892
PVS-6	446	194	640
PVS-7	596	260	856
PVS-8	583	241	824
PVS-9	416	194	610

homogeneous train/validation split. This experiment aimed to evaluate the applicability of our method in a more calibrated scenario.

Finally, alongside the classification experiments, the computational performance of SepRNet-1D, SepSERNet-1D, state-of-the-art architectures, and those proposed by [5] was compared. This analysis validated the effectiveness of the proposed approach and highlighted its potential applicability in environments with limited computational resources.

A. COMPARISON WITH BENCHMARK MODELS

A large number of configurations of the proposed SepRNet-1D were trained and tested to identify the optimal architecture configuration that achieves the best classification performance. To avoid redundant simulations, the modified SepSERNet-1D was trained and tested using the optimal

TABLE 14. Results in terms of classification metrics in the Custom Experiment, for the SepRNet-1D, SepSERNet-1D, the best-performing CNN, LSTM, CNN-LSTM models as well as the other state-of-the-art architectures.

	Accuracy	Precision	Recall	F1-score
SepRNet-1D	96.48	96.15	95.95	96.04
SepSERNet-1D	**97.77**	**97.53**	**97.45**	**97.49**
Best CNN [5]	95.74	95.25	95.09	95.17
Best LSTM [5]	92.54	91.63	91.42	91.52
Best CNN-LSTM [5]	97.04	96.67	96.59	96.63
LeNet [24]	91.57	90.48	90.29	90.34
Xception [27]	91.85	90.96	90.70	90.77
Truncated ResNet18 [26]	92.91	92.00	92.00	92.00
ShuffleNet v2 0.25x [29]	90.78	89.74	89.39	89.48
Tiny ShuffleNet v2 0.25x [29]	88.89	87.47	87.25	87.29
Transformer [30]	94.31	93.79	93.46	93.71
Transformer tiny [30]	93.70	92.86	92.82	92.83
MobileNet V3-small [8]	88.89	87.41	87.23	87.31
Trunc. MobileNet V3-small [8]	93.10	92.20	92.25	92.23

TABLE 15. Results in terms of classification metrics for each category in the Custom Experiment, obtained by SepRNet-1D, SepSERNet-1D, the best-performing CNN, LSTM, and CNN-LSTM models, as well as other state-of-the-art architectures.

	Asphalt road			Cobblestone road			Dirt road		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
SepRNet-1D	98.49	99.89	99.19	95.71	92.65	94.16	94.26	95.32	94.79
SepSERNet-1D	**99.35**	**99.89**	**99.62**	**96.34**	**96.64**	**96.49**	**96.90**	**95.81**	**96.35**
Best CNN [5]	98.7	99.89	99.29	93.26	92.81	93.03	93.79	92.58	93.18
Best LSTM [5]	97.85	99.78	98.81	88.13	86.58	87.35	88.91	87.9	88.4
Best CNN-LSTM [5]	99.35	99.89	99.62	95.34	94.73	95.03	95.31	95.16	95.24
LeNet [24]	98.38	99.89	99.13	88.41	82.91	85.57	84.65	88.06	86.32
Xception [27]	97.42	99.34	98.37	89.67	83.22	86.33	85.78	89.52	87.61
Truncated ResNet18 [26]	98.80	98.80	98.80	88.23	88.66	88.45	88.98	88.55	88.76
ShuffleNet v2 0.25x [29]	97.64	99.67	98.64	83.10	88.02	85.49	88.48	80.48	84.29
Tiny ShuffleNet v2 0.25x [29]	98.06	99.34	98.70	79.93	84.66	82.23	84.41	77.74	80.94
Transformer [30]	97.23	99.78	98.49	92.83	88.98	90.86	91.32	91.61	91.41
Transformer tiny [30]	99.23	99.34	99.29	88.32	90.58	89.48	91.04	88.55	89.78
MobileNet V3-small [8]	97.53	99.56	98.54	81.24	81.63	81.43	83.44	80.48	81.94
Trunc. MobileNet V3-small [8]	99.01	98.58	98.79	88.80	88.66	88.73	88.80	89.52	89.16

architecture configuration found for SepRNet-1D, with the reduction ratio set to $r = 4$.

The architecture parameters were selected using a grid search approach, adjusting the kernel size, the number of internal channels C_{in}, the number of output channels C_{out}, the dropout rate r for each block, and the parameters of the Dense unit block positioned after the GAP layer.

Tab. 3 reports the best configurations of the SepRNet-1D architecture identified through this method.

All the configurations share identical parameters for the initial convolutional layer, including the number of output channels and the kernel size.

Config #7 introduces the block containing a Dropout layer, a Dense layer, ReLU activation, and BatchNormalization. For the remaining configurations, the final Dense layer with Softmax activation was placed directly after the GlobalAveragePooling layer.

Config #10 is similar to config #2, but excludes the SpatialDropout1D layer from both the residual units and the initial convolution block.

For each of the three experiments, the classification results were evaluated using macro-averaged metrics across classes, namely: Accuracy, Precision, Recall and F1-score. The average values across the three experiments were calculated to provide a comprehensive assessment of classification performance, as the experiments use different training and validation data. Tab. 4 presents the classification results in terms of Accuracy, Precision, Recall, and F1-score on the validation set for each experiment, along with the average values across all experiments.

- The configuration of the proposed SepRNet-1D that achieved the best overall performance was config #2, which obtained the highest averaged classification metrics across experiments. Specifically, an average accuracy of 94.69% was achieved.
- Experiment 2 proved to be the most challenging, as all configurations exhibited the lowest performance across the various train/validation splits. This outcome can be attributed to the pronounced mismatch between the training and validation data in this experiment.

Specifically, the validation set contained signals collected from a different vehicle and driver compared to the training set. The dependence of the acceleration response on vehicle properties significantly hindered the models' generalization ability.

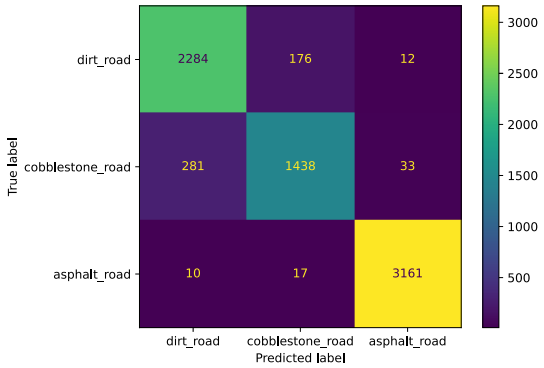
- The highest accuracy values for Experiments 1, 2 and 3 were achieved by config #9, config #8, and config #2, respectively. These configurations shared the same number of blocks, internal channels, and output channels. However, using a kernel size of 3 instead of 7 for the depthwise convolutions maintained good accuracy and F1-score obtained in Experiment 2 but noticeably reduced these metrics in the other splits. The use of a larger number of internal channels, as in config #9 (similar to the residual inverted bottlenecks of MobileNet V3) did not yield an improvement in the averaged metrics.
- Increasing the number of block from 2 to 3 did not result in significant improvement in config #3. Similarly, the addition of a Dense layer, ReLU activation and a Dropout in config #7 led to a decrease in accuracy of 1.18% and 1.61% for Experiments 1 and 2, respectively, compared to config #2.
- The removal of the SpatialDropout1D layer from config #2 resulted in a significant deterioration in performance, as demonstrated by config #10, which exhibited the worst classification metrics among all configurations.

For comparison, Tab. 5, 6, and 7 report the classification results for CNN, LSTM and CNN-LSTM models proposed in [5]. The configurations of the three models (8 CNN, 7 LSTM and 6 CNN-LSTM) are defined in [5]. In this study, the architectures have been implemented, trained, and tested using signals collected from the dashboard, whereas in the reference paper, the models were trained with sensors below the suspension.

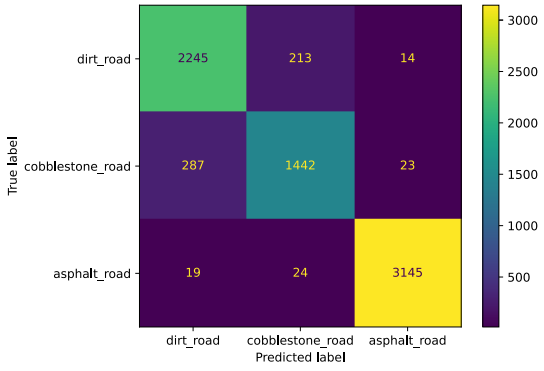
Fig. 7 illustrates two examples of signal frames labeled as asphalt road, which were misclassified by SepRNet-1D as cobblestone road and dirt road, respectively. Fig.8 shows the cumulative confusion matrices for the best-performing CNN, LSTM, and CNN-LSTM models, as proposed in [5]. Conversely, Fig. 9 presents the cumulative confusion matrices for the proposed SepRNet-1D and SepSERNet-1D architectures. The confusion matrices were computed by aggregating all samples from the validation sets across all experiments.

The strong generalization capabilities demonstrated by the proposed architectures SepRNet-1D and SepSERNet-1D architectures across all the validation splits, indicate their adaptability for road type classification using vibrational signals in real-world scenarios.

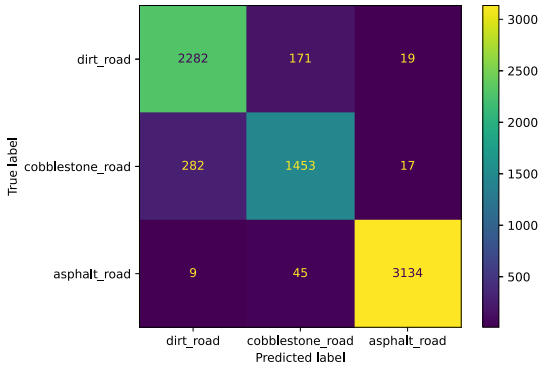
Tab. 8 presents the classification metrics for each experiment related to the configuration of SepRNet-1D, which achieved the best average performance across all experiments among the models whose results are reported in Tab. 4,



(a) Best CNN



(b) Best LSTM



(c) Best CNN-LSTM

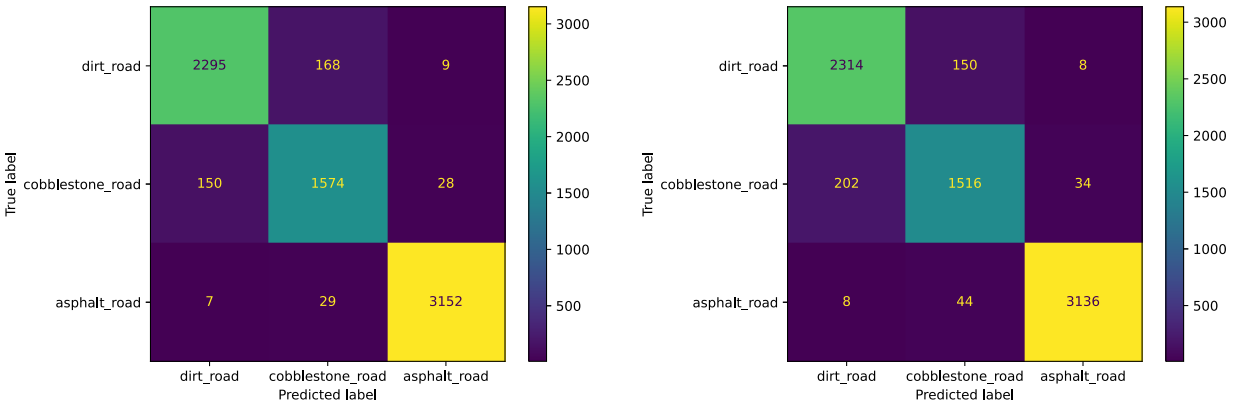
FIGURE 8. Cumulative confusion matrices of best architecture configurations proposed by [5].

and compared to SepSERNet-1D, which shares the same parameter settings.

Tab. 9 reports the classification metrics averaged across all the experiments for each class for best-performing proposed architectures and CNN, LSTM, and CNN-LSTM models proposed in [5].

B. COMPARISON WITH STATE-OF-THE-ART NETWORKS

This section compares the results obtained using SepRNet-1D with those achieved by other widely utilized state-of-the-art CNN architectures.



(a) SepRNet-1D (b) SepSERNet-1D

FIGURE 9. Cumulative confusion matrices of SepRNet-1D and SepRSERNet-1D.

The architectures selected for this comparison include LeNet [24], ResNet-18 [26], ShuffleNet V2 0.25x [29], Xception [27], Transformer [30] and MobileNet V3 [8]. The key innovations and design principles of these architectures were previously analyzed in Subsection II-B, where their unique contributions were also discussed.

However the original designs of these architectures computationally demanding, either due to the complexity of the tasks involved or the optimization of architecture itself. To ensure a fair comparison, lightweight adaptations of these architectures were implemented as follows:

- LeNet, due to of its inherently low computational cost, was considered lightweight in its original form.
- ResNet-18 was adapted by truncating the residual block chain after the first two stages.
- ShuffleNet V2 0.25x was truncated after the first two residual blocks with shuffle units, resulting in a lightweight version referred to as Tiny ShuffleNet V2 0.25x.
- In addition to the original Transformer architecture proposed in [30], a lightweight variant referred to as Transformer-tiny was implemented in this paper. This version reduces both the number of convolutions and the number of transformer encoder blocks to two, while preserving the original number of convolution filters and encoder output dimensions.
- MobileNet V3-small was modified by truncating the backbone after the first two inverted residual bottlenecks.

These networks were trained on the same three splits used for SepRNet-1D and the benchmark models from [5], using identical training hyperparameters, including the optimizer, learning rate, batch size, and stopping condition.

To train the 2D-CNN models, the input signal frame matrix of size $W \times N_{var}$ was treated as an image-like input with dimensions $W \times N_{var} \times 1$. In contrast, the Transformer models process the input matrix with 1D convolutions as a 1D multi-dimensional timeseries with a shape $W \times N_{var}$.

Tab. 10 reports the classification metrics averaged across the experiments for the proposed models and the state-of-the-art models.

SepRNet-1D consistently outperformed all the models across all averaged metrics, demonstrating its effectiveness. LeNet, due to its simplistic design, exhibited the poorest classification performance, achieving only 88.38% accuracy. This reflects its inefficiency in handling too much complex tasks effectively.

Similarly, both ShuffleNet V2 and its lightweight counterpart, Tiny ShuffleNet V2, did not provide satisfactory results, with average accuracies below 89%.

Intermediate performance was observed for the truncated ResNet-18 and Xception architectures. Despite their larger sizes and computational overhead, which typically enhance feature extraction capabilities, these factors may have affected their generalization ability in this context.

The best performance among state-of-the-art models was achieved by the truncated MobileNet V3-small, which demonstrated good generalization capabilities with an averaged accuracy of 93.12%.

The full version of MobileNet V3-small performed significantly worse than its truncated counterpart. This decline in performance may be attributed to its deeper architecture, which likely made it more prone to overfitting.

The original Transformer model proposed in [30] obtained superior performance compared to majority of 2D-CNN models such as LeNet, Truncated ResNet-18, ShuffleNet V2 and Tiny ShuffleNet V2, with an average accuracy of 92.40%. These results highlight the robustness and the adaptability of this architecture for classification tasks involving raw-time domain signals across diverse domains. Despite this, the Transformer did not surpass both the proposed models SepRNet-1D and SepSERNet-1D and also Truncated MobileNetV3small. The lightweight variant, Transformer-tiny, yielded lower performance with a 91.14% average

accuracy. This results suggests that the performance gap is not attributable to overfitting in the full Transformer model, but rather to architectural differences and domain-specific optimization.

In summary, 1D-CNNs and hybrid CNN-LSTM models provided to be much more effective for tasks involving raw time-domain signals compared to 2D-CNNs, which are typically applied to images. The inherent one-dimensional structure of 1D-CNNs and CNN-LSTM architectures indeed makes them significantly more effective than 2D-CNNs in extracting features from raw time-domain signals. The Transformer model, thanks to its design, which include 1D convolutions and self-attention mechanisms, provided comparable results to 1D-CNNs, while not reaching the performance of the proposed models.

The best results were obtained by the SepRNet-1D architecture, which was specifically designed to address the characteristics of vibrational signal data.

This superiority is further supported by Tab. 11, which reports the classification metrics for each road type.

C. CUSTOM EXPERIMENT

As discussed in previous subsections, the three training/validation splits selected for the experiments were intentionally designed to contain unbalanced data to evaluate the effectiveness of the proposed approach under uncalibrated conditions.

Conversely, it is also important to assess the feasibility of the proposed model in a more calibrated scenario, where homogeneous training and validation sets are used.

To investigate the performance of the proposed SepRNet-1D and SepSERNet-1D architectures under balanced conditions, an additional experiment was conducted using balanced training and validation sets.

Signals from all PVS subsets (PVS-1 to PVS-9) were segmented into non-overlapping windows of 300 samples and then randomly split into homogeneous training and validation sets with a 70%/30% ratio. The consistency of the custom-generated split, in terms of the number of samples per class and the distribution of samples across the different PVS subsets, is reported in Tab. 12 and Tab. 13, respectively.

To investigate the classification performance on the validation set of the custom experiment, SepRNet-1D and SepSERNet-1D were compared to the best-performing CNN, LSTM, and CNN-LSTM models. Additionally, the other state-of-the-art architectures were also included in the comparison.

In this experiment, the Adam optimizer was employed with a batch size of 64 and an initial learning rate of 0.001, which was reduced by a factor of 0.8 every 50 epochs without improvement in validation accuracy. Training was terminated after a maximum of 200 epochs without further improvement.

Tab. 14 presents the results in terms of averaged classification metrics across classes on the custom experiment for SepRNet-1D, SepSERNet-1D, the best CNN, LSTM, CNN-LSTM models, as well as the other state-of-the-art architectures.

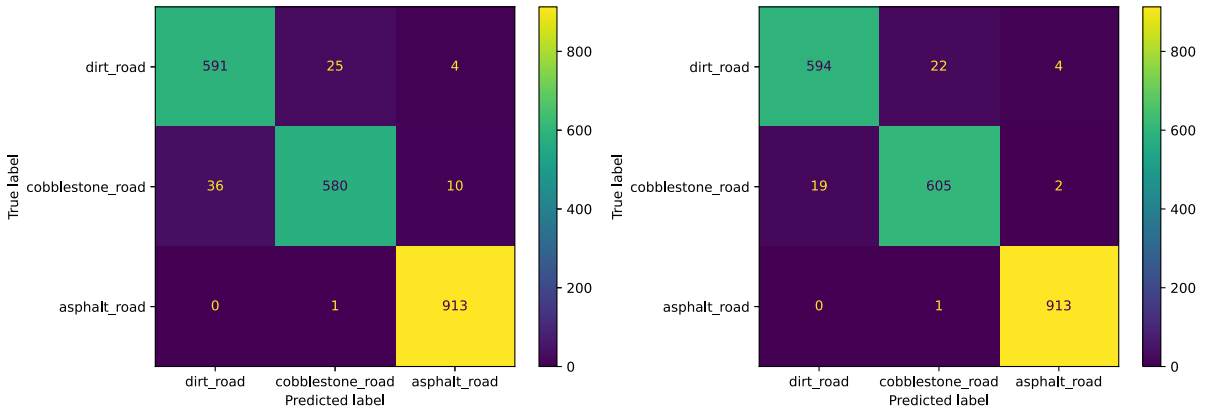
The modified architecture, SepSERNet-1D, achieved the highest classification metrics among all models, with a notable improvement of 1.29% in accuracy compared to SepRNet-1D. These results suggest that the SE module's effectiveness in reducing the less essential features enhances classification performance in scenarios with balanced training and validation sets. Conversely, in the unbalanced sets scenario, the presence of less informative features was already limited, reducing the impact of the SE module's feature recalibration mechanism. As a result, its ability to enhance classification performance was diminished, suggesting that the benefits of channel-wise attention are more pronounced when the feature distribution is more uniform across classes.

Both SepSERNet-1D and SepRNet-1D outperformed the best CNN and LSTM models, with a particularly pronounced performance gap between these two groups. Notably, the CNN-LSTM model demonstrated strong generalization capabilities in this experiment, surpassing both CNN and LSTM models and also slightly outperforming SepRNet-1D. However, SepSERNet-1D achieved the best overall performances, highlighting the effectiveness of the modified architecture in this context.

These findings are further supported by the classification metrics for each class, as reported in Tab. 15.

Fig. 10 illustrates the confusion matrices obtained in the custom experiment by SepRNet-1D and SepSERNet-1D architectures.

A comprehensive analysis of the results across all experiments conducted in this study, including the custom experiment, demonstrates that SepSERNet-1D consistently outperformed all state-of-the-art architectures in both balanced and unbalanced scenarios. For unbalanced sets, SepRNet-1D achieved the highest classification metrics, showcasing its robustness in handling imbalanced data. Conversely, SepSERNet-1D exhibited superior performances for balanced datasets. The strong performance across different scenarios underscore the versatility and effectiveness of SepSERNet-1D in addressing the challenges of road type classification using raw vibrational data. Data dependence on driver style, car type, sensor position in the car, and sensor quality (including bias and white noise affecting the IMU sensor) can negatively impact the accuracy of road type classification. These limitations can be partially mitigated through data normalization during the preprocessing phase, and further improved by appropriately balancing the training and test dataset, as proposed in this custom experiment. The improvement in classification accuracy is supported by the results presented in this section. A comparison between Tables 13 and 14 reveals that, in general, all architectures exhibit enhanced performances in the custom balanced experiment proposed in this work, compared to those proposed by Menegazzo et al. [5]. These limitations can be partially mitigated through data normalization during the



(a) SepRNet-1D (b) SepSERNet-1D

FIGURE 10. Confusion matrices of SepRNet-1D and SepSERNet-1D for the custom experiment.

preprocessing phase, and further improved by appropriately balancing the training and test dataset, as proposed in this custom experiment. The improvement in classification accuracy is supported by the results presented in this section. A comparison between Tables 13 and 14 reveals that, in general, all architectures exhibit enhanced performances in the custom balanced experiment proposed in this work, compared to those proposed by Menegazzo et al. [5].

D. COMPUTATIONAL EXPERIMENTS

In addition to the classification experiments described in the previous subsections, several computational experiments were conducted to evaluate the computational cost, number of parameters, memory usage and inference time for each model on a desktop platform.

To demonstrate the potential feasibility of the proposed model for embedded and mobile system implementations, the computational experiments were further extended to include the memory usage and inference time obtained after converting the models to TensorFlow Lite (.tflite) quantized format. Quantization to the.tflite format, is one of the most widely adopted techniques to accelerate model execution on embedded devices with limited memory and computational capacity.

The proposed SepSERNet-1D, SepRNet-1D, and the other state-of-the-art models were converted from Keras.h5 to.tflite format using float-fallback quantization. This process converts the model weights to 8 bit integers while maintaining activations at 32 bit precision.

Tab. 16 reports the computational results for each model in terms of the number of parameters, computational cost of a single sample inference (measured in Floating Point Operations, FLOPs), memory usage (in MB) for both.h5 and.tflite formats, and inference time per sample for the.h5 format (evaluated on an NVIDIA Tesla T4 GPU) and the.tflite format (evaluated on a CPU with 13 GB RAM). For both formats, inference time was computed by averaging the inference times obtained over 1,000 samples.

The conversion to.tflite format for the best CNN-LSTM model was not feasible due to the Keras TimeDistributed layer which is not supported by the TensorFlow Lite converter.

LeNet, Tiny ShuffleNet V2 0.25x, Truncated MobileNet V3-small, and the best CNN model demonstrated highly competitive computational performance compared to SepRNet-1D and SepSERNet-1D. However, the very low inference time and computational complexity of LeNet and Tiny ShuffleNet V2 0.25x were offset by their poor classification performance.

SepRNet-1D and SepSERNet-1D demonstrated excellent computational performance, achieving inference times of less than 4 ms in.tflite format, while maintaining superior classification quality across all experiments.

Truncated MobileNet V3-small and the best CNN showed slightly better computational performance, with MobileNet V3-small achieving an inference time of 1 ms.

The best CNN model outperformed all the state-of-the-art architectures in balancing computational complexity and classification quality. Nevertheless, SepRNet-1D and SepSERNet-1D provides better classification performance across diverse scenarios while maintaining a strong computational efficiency.

Transformer-tiny model exhibited lower computational efficiency compared to the proposed SepRNet-1D and SepSERNet-1D as well as to the Truncated MobileNet V3-small. In addition, it achieves inferior classification accuracy relative to the proposed models and several other state-of-art architectures.

Finally, the computational performances of the LSTM and CNN-LSTM models (on a desktop GPU) was significantly worse than that of the best CNN model due to the temporal delays introduced by the LSTM units. Furthermore, both Truncated MobileNet V3-small and the best CNN model failed to achieve satisfactory classification accuracy in the balanced datasets scenario.

TABLE 16. Results of the computational experiments for SepSERNet-1D, SepRNet-1D, the best-performing CNN, LSTM and CNN-LSTM models, as well as the other state-of-the-art architectures.

Model	Number of parameters	Computational complexity (MFLOPs)	Memory cost .h5 (.MB)	Inference time on GPU (.h5)	Memory cost .tflite (MB)	Inference time .tflite
SepRNet-1D	79299	45.54	1.07	28.29	0.12	2.64
SepSERNet-1D	89779	45.59	1.23	48.23	0.13	3.47
Best CNN [5]	12355	6.35	**0.21**	**17.75**	**0.02**	0.42
Best LSTM [5]	215603	**0.43**	2.54	1497.34	0.24	49.15
Best CNN-LSTM [5]	356815	0.51	4.20	121.77	-	-
LeNet [24]	1021111	7.76	11.70	6.60	0.98	1.43
Xception [27]	20867051	5202.98	239.10	103.03	20.53	121.83
Truncated ResNet18 [26]	682950	227.06	8.00	22.50	0.68	9.74
ShuffleNet v2 0.25x [29]	109329	11.73	3.08	308.70	0.34	2.65
Tiny ShuffleNet v2 0.25x [29]	**7809**	1.11	0.32	48.44	0.027	**0.33**
Transformer [30]	326403	202.74	4.10	129.38	0.50	21.73
Transformer tiny [30]	134531	69.06	1.70	51.25	0.23	7.32
MobileNet V3-small [8]	1532755	169.82	18.40	173.28	1.64	12.30
Trunc. MobileNet V3-small [8]	7947	4.95	0.28	23.18	0.02	1.12

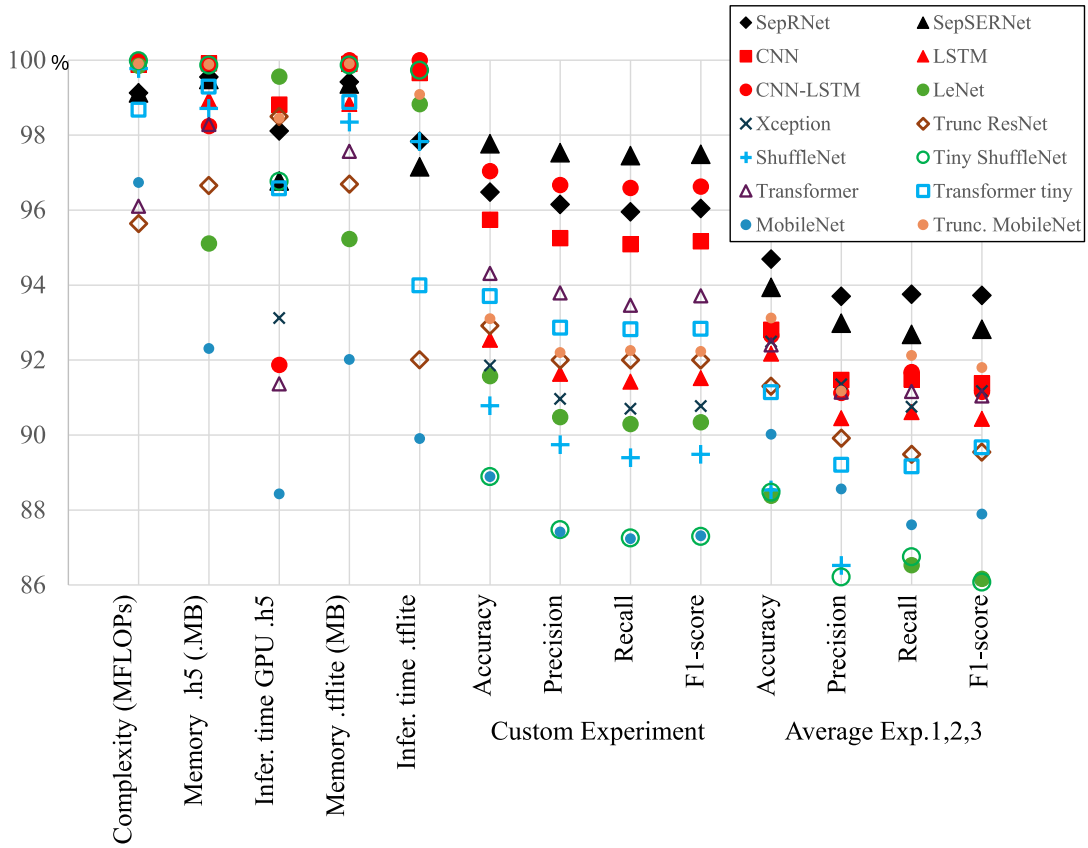


FIGURE 11. Comparison of computational and classification metrics for all the simulated architectures.

Based on these findings, it can be concluded that the proposed models offers a superior trade-off between classification quality across diverse scenarios and computational performance compared to all the state-of-the-art architectures.

Fig. 11 provides a graphical summary of the performances of the 14 architectures considered. The computational performance metrics reported in Tab. 16 were normalized to better visualize the results alongside the classification performance metrics reported in Tab. 11 and Tab. 14.

The normalized performances have been obtained using the following relation $P_{norm} = (1 - (P/P_{max}))$.

In this way the values of the normalized performances varies from 0 to 100% and the best performance corresponds to 100%.

VI. CONCLUSION

This paper addresses the problem of road type classification using deep learning architectures applied to vibrational signals collected from inertial sensors. A total of 46 configurations across 14 architectures were designed, trained and validated to comprehensively evaluate their performance.

In particular, this study proposed SepRNet-1D, a custom, lightweight 1D-CNN architecture that leverages separable convolutions and residual blocks, specifically tailored for road type classification using vibrational signals from inertial sensors. Additionally, a modified version, named SepSERNet-1D, was introduced, incorporating Squeeze-and-Excitation (SE) modules to enhance feature recalibration and improve adaptability.

Comprehensive evaluations were performed using the benchmark dataset [5], employing the same preprocessing steps, training hyperparameters, and training/validation splits as the original benchmark to ensure a fair comparison. Furthermore, a custom experiment with balanced training/validation sets was conducted to assess the feasibility of the proposed models in diverse scenarios.

The results demonstrated that SepRNet-1D and SepSER Net-1D consistently outperformed state-of-the-art architectures and the benchmark models in terms of classification metrics across imbalanced training/validation sets used in the benchmark. For balanced datasets, SepSERNet-1D exhibited improved performance compared to SepRNet-1D, highlighting its adaptability in both calibrated and uncalibrated settings.

Several computational experiments underscored the efficiency of SepRNet-1D and SepSERNet-1D, showcasing competitive computational costs, memory usage, and inference times on a 13 GB RAM desktop CPU after quantization to TensorFlow Lite format.

Future research directions could include further optimization of the architecture by incorporating next-generation residual units based on separable convolutions, such as those utilized in ConvNext [36]. Additionally, the potential benefit of integrating spatial and channel attention modules could be explored.

Finally, the feasibility deploying the proposed architectures for real-time detection on embedded or mobile systems could be demonstrated by evaluating their computational performance on such devices.

REFERENCES

[1] S. Seid, M. Zennaro, M. Libse, and E. Pietrosemoli, "Mobile crowdsensing based road surface monitoring using smartphone vibration sensor and Lorawan," in *Proc. 1st Workshop Experiences Design Implement. Frugal Smart Objects*, Sep. 2020, pp. 36–41.

[2] C. Wu, Z. Wang, S. Hu, J. Lepine, X. Na, D. Ainalis, and M. Stettler, "An automated machine-learning approach for road pothole detection using smartphone sensor data," *Sensors*, vol. 20, no. 19, p. 5564, Sep. 2020.

[3] B. Varona, A. Monteserin, and A. Teyseyre, "A deep learning approach to automatic road surface monitoring and pothole detection," *Pers. Ubiquitous Comput.*, vol. 24, no. 4, pp. 519–534, Aug. 2020.

[4] P. Singh, A. E. Kamal, A. Bansal, and S. Kumar, "Road pothole detection from smartphone sensor data using improved LSTM," *Multimedia Tools Appl.*, vol. 83, no. 9, pp. 26009–26030, Aug. 2023.

[5] J. Menegazzo and A. von Wangenheim, "Road surface type classification based on inertial sensors and machine learning," *Computing*, vol. 103, no. 10, pp. 2143–2170, Oct. 2021.

[6] L. Manoni, S. Orcioni, and M. Conti, "Recent advancements in deep learning techniques for road condition monitoring: A comprehensive review," *IEEE Access*, vol. 12, pp. 154271–154293, 2024.

[7] E. A. Martinez-Ríos, M. R. Bustamante-Bello, and L. A. Arce-Sáenz, "A review of road surface anomaly detection and classification systems based on vibration-based techniques," *Appl. Sci.*, vol. 12, no. 19, p. 9413, Sep. 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/19/9413

[8] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.

[9] O. A. Egaji, G. Evans, M. G. Griffiths, and G. Islas, "Real-time machine learning-based approach for pothole detection," *Expert Syst. Appl.*, vol. 184, Dec. 2021, Art. no. 115562.

[10] A. Martinelli, M. Meocci, M. Dolfi, V. Branzi, S. Morosi, F. Argenti, L. Berzi, and T. Consumi, "Road surface anomaly assessment using low-cost accelerometers: A machine learning approach," *Sensors*, vol. 22, no. 10, p. 3788, May 2022.

[11] A. Basavaraju, J. Du, F. Zhou, and J. Ji, "A machine learning approach to road surface anomaly assessment using smartphone sensors," *IEEE Sensors J.*, vol. 20, no. 5, pp. 2635–2647, Mar. 2020.

[12] I. Ferjani and S. Ali Alsaif, "How to get best predictions for road monitoring using machine learning techniques," *PeerJ Comput. Sci.*, vol. 8, p. e941, Apr. 2022.

[13] A. Salman and A. N. Mian, "Deep learning based speed bumps detection and characterization using smartphone sensors," *Pervas. Mobile Comput.*, vol. 92, May 2023, Art. no. 101805.

[14] A. Sabapathy and A. Biswas, "Road surface classification using accelerometer and speed data: Evaluation of a convolutional neural network model," *Neural Comput. Appl.*, vol. 35, no. 19, pp. 14183–14194, Jul. 2023.

[15] I. Siddiqui, S. Mazhar, N. Hassan, and W. Sultani, "Fine-grained road quality monitoring using deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 10, pp. 10691–10701, Oct. 2023.

[16] E. A. Martinez-Ríos, R. Bustamante-Bello, and S. A. Navarro-Tuch, "Generalized Morse wavelets parameter selection and transfer learning for pavement transverse cracking detection," *Eng. Appl. Artif. Intell.*, vol. 123, Aug. 2023, Art. no. 106355.

[17] S. Tiwari, R. Bhandari, and B. Raman, "RoadCare: A deep-learning based approach to quantifying road surface quality," in *Proc. 3rd ACM SIGCAS Conf. Comput. Sustain. Societies*, Jun. 2020, pp. 231–242.

[18] G. Baldini, R. Giuliani, and F. Geib, "On the application of time frequency convolutional neural networks to road Anomalies' identification with accelerometers and gyroscopes," *Sensors*, vol. 20, no. 22, p. 6425, Nov. 2020.

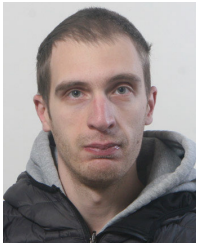
[19] E. Raslan, M. F. Alrahmawy, Y. A. Mohammed, and A. S. Tolba, "IoT for measuring road network quality index," *Neural Comput. Appl.*, vol. 35, no. 3, pp. 2927–2944, Jan. 2023.

[20] A. K. Pandey, R. Iqbal, T. Maniak, C. Karyotis, S. Akuma, and V. Palade, "Convolution neural networks for pothole detection of critical road infrastructure," *Comput. Electr. Eng.*, vol. 99, Apr. 2022, Art. no. 107725.

[21] H. Xin, Y. Ye, X. Na, H. Hu, G. Wang, C. Wu, and S. Hu, "Sustainable road pothole detection: A crowdsourcing based multi-sensors fusion approach," *Sustainability*, vol. 15, no. 8, p. 6610, Apr. 2023.

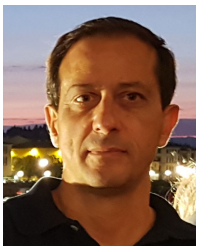
[22] T. Dózsa, V. Jurdana, S. B. Šegota, J. Volk, J. Radó, A. Soumelidis, and P. Kovács, "Road type classification using time-frequency representations of tire sensor signals," *IEEE Access*, vol. 12, pp. 53361–53372, 2024.

[23] N. Hnoohom, S. Mekruksavanich, and A. Jitpattanakul, "A comprehensive evaluation of state-of-the-art deep learning models for road surface type classification," *Intell. Autom. Soft Comput.*, vol. 37, no. 2, pp. 1275–1291, 2023.
[24] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[25] S. M. Shahid, S. Ko, and S. Kwon, "Real-time abnormality detection and classification in diesel engine operations with convolutional neural network," *Expert Syst. Appl.*, vol. 192, Apr. 2022, Art. no. 116233.
[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[27] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.
[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[29] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet v2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 122–138.
[30] Y. Shavit and I. Klein, "Boosting inertial-based human activity recognition with transformers," *IEEE Access*, vol. 9, pp. 53540–53547, 2021.
[31] M. Hussain, C. Cheng, R. Xu, and M. Afzal, "CNN-fusion: An effective and lightweight phishing detection method based on multi-variant ConvNet," *Inf. Sci.*, vol. 631, pp. 328–345, Jun. 2023.
[32] X. Li, X. Kong, Z. Liu, Z. Hu, and C. Shi, "A novel framework for early pitting fault diagnosis of rotating machinery based on dilated CNN combined with spatial dropout," *IEEE Access*, vol. 9, pp. 29243–29252, 2021.
[33] W. Gong, H. Chen, Z. Zhang, M. Zhang, and H. Gao, "A data-driven-based fault diagnosis approach for electrical power DC–DC inverter by using modified convolutional neural network with global average pooling and 2-D feature image," *IEEE Access*, vol. 8, pp. 73677–73697, 2020.
[34] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
[35] J. Menegazzo and A. von Wangenheim, "Speed bump detection through inertial sensors and deep learning in a multi-contextual analysis," *Social Netw. Comput. Sci.*, vol. 4, no. 1, p. 18, Oct. 2022.
[36] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11966–11976.



LORENZO MANONI (Member, IEEE) received the B.Sc. and M.Sc. degrees in electronics engineering and the Ph.D. degree in information engineering from the Università Politecnica delle Marche, Ancona, Italy, in 2015, 2018, and, 2022, respectively. He was a Research Fellow with the Department of Information Engineering (DII), in 2022. He is currently a Researcher with DII. His current research interests include embedded systems, machine learning, deep learning, artificial intelligence applications, convolutional neural networks, signal processing, algorithms analysis and design, and bio-signal analysis.

SIMONE ORCIONI (Senior Member, IEEE) received the Laurea and Ph.D. degrees in electronics engineering from the Universitá Politecnica delle Marche, Ancona, Italy, in 1992 and 1995, respectively. He was appointed as an Assistant Professor, in 2000, teaching courses in analog and digital electronics and authored a text book. In 2017, he was an Adjunct Professor with the Ubiquitous Computing Laboratory (UC-Lab), HTWG Konstanz, Germany, where he is currently a Guest Researcher. Since 2021, he has been held the position of an Associate Professor with the Department of Information Engineering, Universitá Politecnica delle Marche, where he is also the President of the Board of the Degree Programs in Electronic Engineering. He has authored more than 50 journal articles and over 100 contributions to international conference proceedings and edited book chapters. His research has addressed statistical device modeling and simulation, analog circuit design, cyber-physical system simulation, and both linear and nonlinear system identification. His current research interests include nonlinear digital signal processing, li-ion battery impedance spectroscopy, and electronics for renewable energy applications. He has acted as a reviewer for 20 international journals, a program committee member for seven international conferences, the program chair for three international conferences, an editor for four international volumes, and an inventor for two patents. He has been included in Stanford University's "World's Top 2% Scientists" in 2023 and 2024 edition of "Annual Influence Ranking" and the "Lifetime Scientific Influence Ranking." He has served as a Guest Editor for *EURASIP Journal on Embedded Systems*, *Frontiers in Energy Research*, and *Sensors* (MDPI).



MASSIMO CONTI (Member, IEEE) received the degree in electronics engineering from the University of Ancona, Italy, in 1987. He is an Associate Professor with the Dipartimento di Ingegneria dell'Informazione (DII), Universitá Politecnica delle Marche (UNIVPM), Ancona, Italy. His research activity in the field of microelectronics was mainly devoted to system level design of low power integrated circuits, electronic smart systems for ambient assisted living, design of energy harvesting systems, battery management systems, vehicle to grid (V2G) connection, smart grids, state of health estimation, analysis of cell mismatch on battery pack performances, design of BMS and battery life tracing for reuse, recycle and end-of-life, smart mobility, and NFC for food traceability. He is the co-author of more than 250 papers on international books, journals, and conferences. He has 190 Scopus publications, 1378 citations, and an h-index of 18. He is a coordinator of European and national research projects. He is an editor of 11 international books. He is a lead guest editor of special issue of international journals. He is the general chairperson of ten international conferences. Further information on www.univpm.it/massimo.conti