Proceedings of the
13th International Conference on Applied Informatics
Eger, Hungary, February 19–21, 2026
DOI: 10.17048/icai.2026.216
pp. 216–226

Acoustic Traffic Monitoring with MEMS Microphones, Concatenated log-Bark Spectrograms and CNN

István Pintér, Lóránt Kovács

John von Neumann University,
GAMF Faculty of Engineering and Computer Science
pinter.istvan@nje.hu, kovacs.lorant@nje.hu

Abstract. The paper presents our results in algorithm-development and solutions for two ATM problems using MEMS microphone-signals. The aim was that the algorithms have to be realised in the sensor itself, thus we followed the edge-AI system-concept. The IDMT-Traffic dataset was used in algorithm development. The paper proposes the concatenated log-Bark spectrogram as feature extraction, followed by the custom CNN classifier. In the motion's direction determination (from left/from right/no vehicle) the overall accuracy was 97.70±0.64%. The detection of passing vehicle in background noise was considered as a two-class classification problem (vehicle/no vehicle). The accuracy, F1-score and Matthew's Correlation Coefficient (MCC) performance parameters achieved were 99.75%, 0.9975, 0.9949, respectively. The conclusion, based on experimental evidence, was that the MEMS-microphones could be used to develop edge-AI type sensors for motion's direction determination and detection of passing vehicle in background noise using the concatenated log-Bark spectrograms converted to grayscale images as input to a CNN-classifier.

Keywords: MEMS-microphone, log-Bark-spectrogram, CNN, motion-direction, passing vehicle detection, accuracy, F1-score, MCC

2020 *Mathematics Subject Classification:* Primary 68T07; Secondary 94A12

1. Introduction

Recent advances in info-communication technologies highlight the concept of smart cities. One objective is the optimal control of urban and suburban vehicle traffic, which requires traffic monitoring. It's main problems are the following: 1) detection and counting of passing vehicles, 2) determination of vehicle's category, 3) determination of motion's direction, and 4) vehicle's speed estimation. The real-time data necessary for traffic monitoring are supplied by appropriate sensors. Inductive loops and pneumatic pipes are highly reliable ones for solving problem 1) and 3). Unfortunately, these are not suitable for real-time traffic monitoring: inductive loops must be built into the road during road-building, and pneumatic pipes must be laid across the road, thus affecting the traffic itself. Another family of sensors are based on electromagnetic operating principles (e.g. radar, lidar, and cameras). However, their – otherwise excellent – detection accuracy is highly dependent on meteorological conditions and time of day [8]. Besides, the high cost of installation and maintenance also calls for another or additional sensor-domain, namely acoustic sensors. This leads to the concept of Acoustic Traffic Monitoring (ATM). Most frequently, pressure-sensing microphones are used, and a promising goal is developing sensors with low-cost, medium-quality omnidirectional MEMS (Micro-ElectroMechanical Systems) microphones. The main efforts in this sensor-development field can be summarized as: (I) minimizing the cost of sensors, (II) decreasing the installation and maintenance costs, (III) decreasing the electrical power consumption of the sensor and signal processing units, (IV) optimizing the suitable sensor network architecture for covering a large geographic area with many cooperating sensor units, and (V) guaranteeing the continuous operation even in adverse weather conditions.

In this paper we present our results in algorithm-development for ATM problems 1) and 3) using MEMS microphone-signals. The problem 2) was also ivestigated. Our aim was that the algorithms have to be realised in the sensor itself, thus we followed the edge-AI system-concept. Several freely available audio datasets exist for this purpose [1, 7, 9]. We used the IDMT-Traffic dataset, because it consists of two-channel audio recordings both with high-quality sE8 and MEMS microphones. Therefore, the performance of MEMS-based algorithms can be compared with sE8-based ones. We considered [1] as reference, because the authors presented possible solutions for problems 1), 2) and 3) in case of sE8 microphones. In solving the problem 3) their neural network was a variant of MobileNet, adapted to the problem at hand. The input was the cross-correlogram, computed as a series of cross-correlations between the two microphone-channels. For problem 2) their classifier was a pretrained CNN (Convolutional Neural Network), which was adapted to the concrete problem using transfer learning. The input was the log-mel spectrogram computed from the average of the two channels' signals.

The structure of the paper is the following. Section 2 summarizes the main properties of the dataset used for algorithm development. Section 3 introduces the concept of concatenated log-Bark features and describes the CNN structure. In Section 4 the numerical results achieved in vehicle category classification can be found, which is followed by in Section 5 and 6 with results of direction of movement classification and performance parameters of passing vehicle detection. The paper ends with conlusions.

2. The IDMT-Traffic dataset

In the work we used the IDMT-Traffic dataset [1], which consists of 17506 two-channel (stereo) wav-files, recorded both with high quality sE8 and medium quality, low cost MEMS microphones. The duration of a particular recording is $t_d = 2\,\mathrm{s}$, the sampling frequency is $f_s = 48\,\mathrm{kHz}$. The dataset contains two main sound categories: background noise (8144 recordings) and sound of passing vehicles (9362). In the latter case there are four sub-categories: car (7804), bus (106), truck (1022) and motorcycle (430). The sounds were recorded in four places (three urban and one country-road) with different weather conditions (dry and wet road) and with different speed limits (30 km/h, 50 km/h and 70 km/h). The direction of motion was also recorded. The recordings are annotated, the audio file-name contains the labels: date, location, speed limit, daytime, weather condition, vehicle category/background noise, motion's direction and microphone type.

3. Concatenated log-Bark features and the CNN

In the recent, relevant literature the input of the classifier is the so-called log-mel spectrogram [1, 7, 9]. The log-mel spectrogram is computed from linear spectrogram using the mel-frequency scale and triangle-shaped, overlapped filters [5]. The mel frequency-scale is a characteristics of human hearing, and its non-linear function is the result of function-fitting to experimental, psychoacoustical data [10]. Another psychoacoustical frequency scale is the Bark-scale, with 24 critical bands of human hearing [12]. Therefore, in an earlier preliminary study – with the subject of detecting passing vehicles in background noise – we compared the effect of the mel/Bark scale on detection accuracy. Beyond that, in that comparison we used both triangular-shaped filters and raised cosine filters, too. The final result of the study was, that in case of sE8 microphone the best combination is mel-scale with raised-cosine filters, and in case of MEMS microphone the Bark-scale with raised cosine filters gave the best accuracy. Therefore, in this work we used the log-Bark spectrogram computed with raised cosine filters. Moreover, in order to preserve the time-delay information between the two microphone channels, we concatenated the log-Bark spectrograms of the two channels ensuring the proper time-alignment. Thus, the concatenated log-Bark spectrogram was the result of feature extraction.

3.1. The concatenated log-Bark spectrogram

The signal was processed using sliding windows with fixed window length ($t_w = 100\,\mathrm{ms}$) and hop ($t_h = 25\,\mathrm{ms}$). Thus the number of samples in a window was $N = f_s \cdot t_w = 4800$, and the number consecutive of windows was $M = t_d/t_h = 80$ (not 77, because the last three windows were zero-padded at the end). In the mth window the signal's one-sided power spectrum $|S_m(l)|^2, m = 1..M, l = 1..N/2$ was estimated using Hamming-window and discrete Fourier-transform (The frequency resolution was $\Delta f = 1/t_w = 10\,\text{Hz}$). The linear spectrogram was a series of the one-sided power spectra.

We processed the linear spectrogram further using a psychoacoustical model of human hearing. From the available non-linear functions (scales) we used the Bark-scale, mentioned above, with the invertible function below [11] :

$$\eta(f) = 6 \cdot \operatorname{asinh}\left(\frac{f[Hz]}{600[Hz]}\right)$$

We covered the $[0; \eta(f_{\max})]$ interval with K overlapped, raised-cosine-type weighting functions (sub-band filters). In these bands we summed up the weighted power values and converted the result into dB. Thus finally we got the log-Bark spectrogram as signal representation of the channel's signal.

The sub-band filters were shifted versions of the raised-cosine prototype function with finite support of $2\Delta\eta$, where $\Delta\eta = \eta(F_s/2)/(K+1)$. The kth sub-band filter therefore:

$$\Psi_k(\eta) = \begin{cases} \frac{1+\cos\left[\frac{\pi\cdot(\eta - k\Delta\eta)}{\Delta\eta}\right]}{2} & \eta_1 \le \eta \le \eta_2 \\ 0 & \text{else} \end{cases}$$

where $\eta_1 = (k-1)\Delta\eta$ and $\eta_2 = (k+1)\Delta\eta$. With the inverse η^{-1} of $\eta(f)$ the corresponding frequency values were $f_1 = \eta^{-1}(\eta_1)$ and $f_2 = \eta^{-1}(\eta_2)$ and the frequency band was $f_1 \le f \le f_2$. The kth sub-band filter in the frequency domain was $H_k(f) = \Psi_k(\eta^{-1}), f \in [f_1; f_2]$. With the frequency resolution above finally we got $H_k(l) = H_k(l \cdot \Delta f), l \in [l_1; l_2]$ where $l_1 = \lfloor f_1/\Delta f \rfloor$, $l_2 = \lceil f_2/\Delta f \rceil$, the sub-band filter in the linear frequency domain for discrete-time processing.

For better visibility Figure 1 shows a raised-cosine filterbank with $K = 3$ filters transformed back to linear frequency scale ($f_s = 8$ kHz, $f_{max} = f_s/2$)

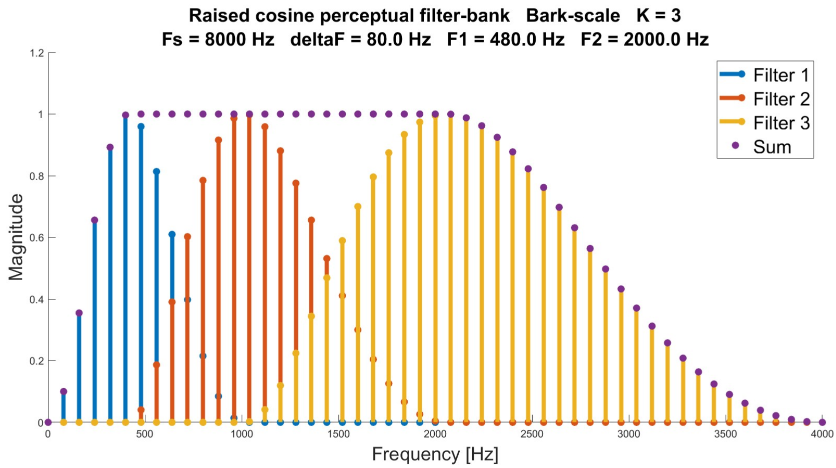

Figure 1. Raised-cosine filterbank example.

The final step in feature extraction was computing the weighted sums from power spectrum in the mth signal window:

$$P_{k,m} = 10 \lg \left(\sum_{l=l_1}^{l_2} |H_k(l)|^2 \cdot |S_m(l)|^2 \right), \quad k = 1..K, \quad m = 1..M$$

The $P_{k,m}$ is the log-Bark spectrogram, which is an $K \times M$ matrix of real values.

In Figure 2 an example can be seen using a recording from the IDMT-Traffic datset. The upper trace is trace 1, which shows the Hamming-windowed 100 ms long signal segment (of 4800 samples). Trace 2 is the one-sided power spectrum (of 2400 components). Trace 3 shows the 24 raised-cosine-type filters, designed in the Bark-scale and warped back to linear frequency scale. Finally in trace 4 the results of weighted sum of power values of each filters can be seen expressed in dB. These computations are resulted in a 24 dimensional vector, which is a column of the log-Bark spectrogram matrix. It is important to note, that changing over to log-Bark spectrogram from linear spectrogram results in a 100-times data reduction.

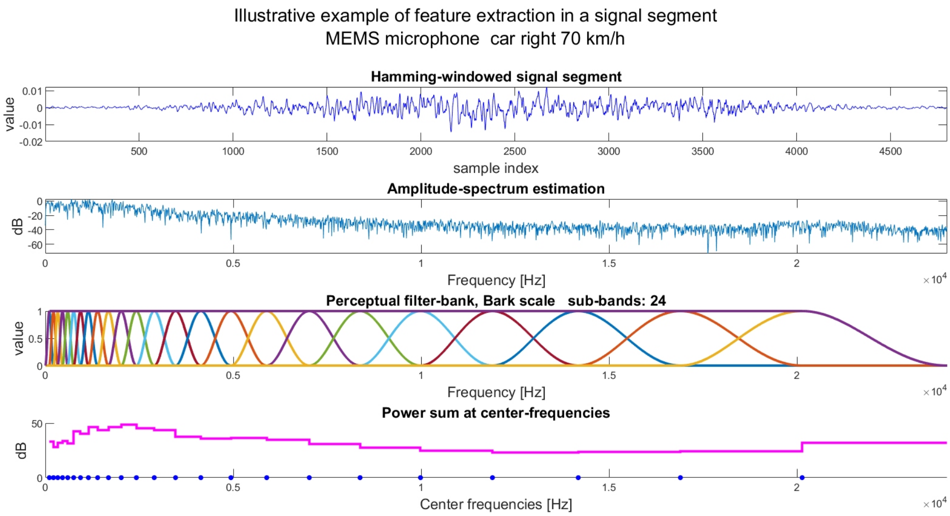

Figure 2. Log-Bark power spectrum of a signal segment.

Figure 3 illustrates that the log-Bark spectrogram representation enhances the weighted power sums in the lower frequency bands. In the figure the log-Bark spectrogram (left) and the linear spectrogram (right) can be seen in case of 50 km/h bus recording with MEMS microphones. The concatenation operation means that the two channels' log-Bark spectrograms are stacked above each other. It ensures the proper time-alignment of the channels' samples. The final result of the feature extraction is thus a 48×80 matrix of real values.

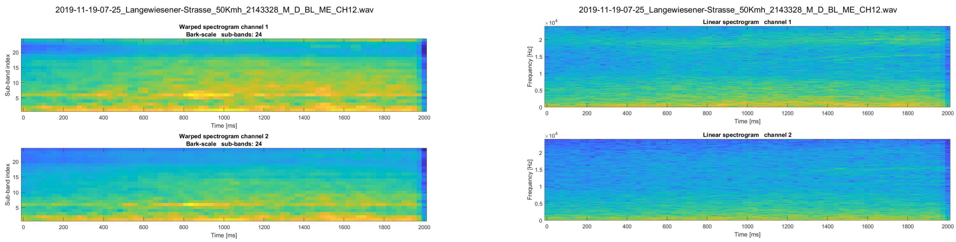

Figure 3. Log-Bark (left) and linear spectrograms (right).

3.2. The CNN structure

The concatenated log-Bark spectrogram representation, described above, can easily be converted to a 48×80 gray-scale image (PNG file-format was used). The interval of the lowest and greatest matrix-elements was scaled to [0;255] of real values linearly, then uniformly quantized to [0;255] interval of nonnegative integers. So it was obvious to select CNN [2] as the classifier part of the system, because it is well-proven and widely-used in ATM problems [1, 7, 9]. However, we intentionally did not use a pre-trained version. It makes it possible to optimize further the network structure if necessary when implementing the CNN classifier in the sensor. In designing the CNN structure we used the MATLAB Deep Learning Toolbox. Interestingly, the CNN processing pipeline was the same in both vehicle category classification and direction of movement determination. The network structure was the following (reLU denotes the $f(x) = \max(0, x)$ function). There were three main building block types. We denote here with type-1 a sequence of <2D convolution layer, batch normalization, reLU activations, max-pooling> operations, with type-2 the <2D convolution layer, batch normalization, reLU activations> sequence and type-3 the <fully conncted layer, softmax activations, classification layer>. Using these notations the image input layer was followed by two type-1 layers. The outputs the second were further processed with a type-2 block. It was followed by a type-3 block for outputting the predicted class labels. The size of convolution layers' filters were 3×3 in each blocks, the number of filters were 8, 16, 32, respectively. Not only the weights, but biases were trained as well. The total number of learnable parameters were 26011. The network was trained using the cross-entropy loss-function and Adam optimizer, with minibatch-size of 128 and initial learning rate of 0.001.

4. Vehicle category classification

From the IDMT dataset we created the "commercial vehicle" class from recordings of trucks and buses in order to distinguishing them from passenger cars. When creating the training/validation and test subsets of recordings we followed the policy of [1]: recordings labelled with 30 km/h and 50 km/h were the training/validation sets and recordings with 70 km/h speed limit was the test set. As a consequence, we exluded all motorcycle recordings, because there are 5 recordings with 30 km/h labels, there is no with 50 km/h and 70 km/h and there are 123 with "Unknown" speed labels. The vehicle category classification was a 3-class classification problem: passenger car (car), commercial vehicle (bus or truck), no vehicle (labelled as BG, background noise in the dataset). We used the `splitEachLabel` function of MATLAB Deep Learning Toolbox with parameters of 0.9 and 'randomize' with default seed. The number of recordings of each set can be found in Table 1 (with 90% training, 10% validation ratios).

Table 1. Number of recordings in datasets.

vehicle category	training	validation	test
car	2471	274	1157
bus or truck	332	37	195
no vehicle	2395	266	1412

The structure of CNN classifier was described in the Subection 3.2. We did 10 training experiments, evaluated the confusion matrix, selected the best network with greatest overall accuracy and computed the average and standard deviation of experiments' accuracies. Figure 4 shows randomly selected training grayscale image examples.

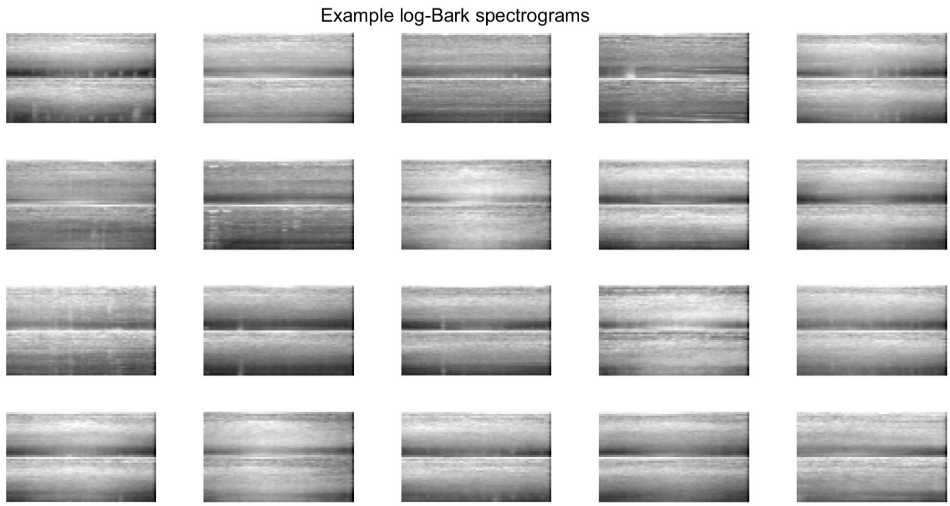

Figure 4. Randomly selected training images for vehicle category classification.

The best confusion matrix in case of 10 experiments and 10 epoch/experiments can be seen in Table 2. The best overall accuracy was 94.79%, the average accuracy was $94.04 \pm 0.48\%$.

Table 2. Vehicle category classification: best confusion matrix.

True label	bus or truck	87 (44.62%)	107 (54.87%)	1 (0.51%)
	car	27 (2.33%)	1128 (97.49%)	2 (0.17%)
	no vehicle	4 (0.28%)	3 (0.21%)	1405 (99.5%)
		bus or truck	car	no vehicle
			Predicted label	

In the reference paper [1] high quality sE8 microphone recordings were used for training a 4-class vehicle category classifiier (car/truck/motorcycle/no vehicle). The corresponding classification ratios are the following: a) car/car decision 97.29% (sE8); 97.49% (MEMS), b) no vehicle/no vehicle decision 99.65% (sE8); 99.5% (MEMS). Let's mention, that the truck/truck ratio was 38.84% (sE8) and the bus or truck/bus or truck ratio was 44.62% (MEMS). The greatest misclassification ratio in case of sE8 was the truck/car with 60.21%, and in case of MEMS the bus or truck/car ratio was 54.87%. To our opinion the primary cause of these high misclassification rates was the underrepresentation of training recordings of trucks (322, sE8) and bus or truck (369, MEMS) compared with cars' (2745). We concluded, that both in case of sE8 and MEMS microphones further data collection is necessary to get a more balanced datasets.

5. Direction of movement classification

In organizing the computations in this case we followed similar methodology as in the vehicle category classification: the motorcycle recordings were excluded, the recordings labelled with 30 km/h or 50 km/h speed limits formed the trainig/validation substes with 90%/10% splits, and the recordings labelled with 70 km/h were the test set. We used the `splitEachLabel` function of MATLAB Deep Learning Toolbox with parameters of 0.9 and 'randomize' with default seed in this case also. Thus, the direction of movement classification was also a 3-class classification problem: from left to right/from right to left/no vehicle. The number of recordings of each set can be found in Table 3.

Table 3. Number of recordings in datasets.

direction of movement	training	validation	test
from left to right	1391	155	635
from right to left	1411	157	717
no vehicle	2395	266	1412

We also did 10 training experiments, selected the best network with greatest overall accuracy and characterized the training process itself with the average and standard deviation of experiments' accuracies. The best confusion matrix in case of 10 experiments and 10 epoch/experiments can be seen in Table 4. The best overall accuracy was 98.63%, the average accuracy was $97.70 \pm 0.64\%$.

Table 4. Direction of movement: best confusion matrix.

	from left to right	616 (97.01%)	16 (2.52%)	3 (0.47%)
True label	from right to left	15 (2.09%)	701 (97.77%)	1 (0.14%)
	no vehicle	3 (0.21%)	0 (0%)	1409 (99.79%)
		from left to right	from right to left	no vehicle
			Predicted label	

In the reference paper [1] the datasets used for training were slightly different from ours. However, some comparison can be made based on confusion matrices. The corresponding classification ratios are the following: a) from left to right/from left to right 96.61% (sE8); 97.01% (MEMS), b) from right to left/from right to left 98.64% (sE8); 97.77% (MEMS) and c) no vehicle/no vehicle 99.79% (sE8) 97.79% (MEMS). We concluded, that MEMS microphone recordings can be used for direction of movement classification with similar performance of sE8.

6. Detection of passing vehicles

The detection of passing vehicles in a background noise is a 2-class classification problem (vehicle/no vehicle). The confusion matrix CM_2 can be derived from the 3-class one (CM_3) as:

$$
\begin{aligned}
CM_2(1,1) &= CM_3(1,1) + CM_3(1,2) + CM_3(2,1) + CM_3(2,2),\\
CM_2(1,2) &= CM_3(1,3) + CM_3(2,3),\\
CM_2(2,1) &= CM_3(3,1) + CM_3(3,2),\\
CM_2(2,2) &= CM_3(3,3).
\end{aligned}
$$

Table 5 shows the confusion matrix with conventional notations.

Table 5. Two-class confusion matrix (CM_2) structure.

True label	vehicle	TP	FN
	no vehicle	FP	TN
		vehicle	no vehicle
		Predicted label	

For numerical assessment of detection we used the following performance parameters (for MCC se [3]):

$$\text{accuracy} = (TP + TN)/(TP + FN + FP + TN)$$

$$F1_{\text{score}} = 2TP/(2TP + FP + FN)$$
$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

The numerical values derived from vehicle category classifier and direction of movement estimator can be seen in Table 6.

Table 6. Performance parameters of vehicle detection.

task	accuracy	F1-score	MCC
vehicle category	99.64%	0.9965	0.9928
direction of movement	99.75%	0.9975	0.9949

The accuracy of motion's direction classifier (99.75%) was somewhat greater, than that of the vehicle category classifier.

In the reference paper [1] the high quality sE8 microphone recordings were used, and there is no explicit acccuracy data available for vehicle/no vehicle detection. However, we can compare the no vehicle/no vehicle decision ratios of corresponding classification tasks. Their TABLE III is the confusion matrix of vehicle category classifier; the ratio of no vehicle/no vehicle decision was 99.65%. TABLE V is the confusion matrix of vehicle category classifier with 99.79%. Our best ratio of 99.79% (see Table 4) was the same as the best reference data (99.79%). Thus the vehicle/no vehicle classification can be solved using our MEMS-based motion's direction classifier with accuracy of 99.75% (see Table 6).

7. Conclusions

Based on the numerical experimental evidence presented in the paper we concluded, that the MEMS-microphones can be used for vehicle/non vehicle classification and direction of movement estimation with acceptable accuracy. The great data reduction with log-Bark spectrogram signal representation and the small number of learnable parameters of the proposed CNN makes it possible to develope an edge-AI type sensor, too. In this respect we will consider the results published in [4, 6].

Another interesting work planned is the comparison the log-Bark representation-based solution with others (e.g. log-mel, MFCC, Gammatone Frequency Cepstral Coefficients, cross-correlogram and Generalized Cross-Correlation with Phase-Transform) using larger datasets of [7, 9] with ablation studies.

However, in the vehicle category determination problem further data collection is needed in case of commercial vehicles in order to get a more balanced training dataset. The poor recognition accuracy caused by the class imbalance is planned to inrease using class weighting and data augmentation.

Acknowledgements

The authors would like to thank the anonymous reviewers. Their comments and suggestions significantly improved the quality of the paper.

References

[1] J. Abesser, S. Gourishetti, et al.: *IDMT-Traffic: An Open Benchmark Dataset for Acoustic Traffic Monitoring Research*, in: Proceedings of 29th EUSIPCO European Signal Processing Conference, Dublin, Ireland, 2021, pp. 551–555, doi: 10.23919/EUSIPCO54536.2021.9616080.

[2] C. M. Bishop, H. Bishop: *Deep Learning, Foundations and Concepts*, Springer Cham, 2024, doi: 10.1007/978-3-031-45468-4.

[3] D. Chicco, G. Jurman: *The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification*, BioData Mining 16.4 (2023), pp. 1–23, doi: 10.1186/s13040-023-00322-4.

[4] R. Cordova-Cardenas et al.: *Edge AI in Practice: A Survey and Deployment Framework For Neural Networks on Embedded Systems*, Electronics 14.4877 (2025), pp. 1–39, doi: 10.3390/electronics14244877.

[5] S. Davis, P. Mermelstein: *Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences*, IEEE Transactions on Acoustics, Speech, and Signal Processing 28.4 (1980), pp. 357–366, doi: 10.1109/TASSP.1980.1163420.

[6] V. Fridous et al.: *Optimization and Bechmarcking of Lightweight Neural Networks for Efficient Embedded AI Deployment*, Engineering Reports 8.e70818 (2026), pp. 1–14, doi: 10.1002/eng2.70818.

[7] S. Ghaffarzadegan et al.: *Sound of Traffic: A Dataset for Acoustic Traffic Identification and Counting*, in: Proceedings of Interspeech 2024, Kos, Greece, 2024, pp. 117–121, doi: 10.21437/Interspeech.2024-1205.

[8] T. Kovács, L. Kovács: *Economical Camera-based Measuring Scheme for Roundabout Traffic*, Periodica Polytechnica Transportation Engineering 53.2 (2025), pp. 235–241, doi: 10.3311/PPtr.37617.

[9] H. Parineh, M. Sarvi, A. B: *MELAUDIS: A Large-Scale Benchmark Acoustic Dataset For Intelligent Transportation Systems Research*, Sci Data 12.362 (2025), pp. 1–13, doi: 10.1038/s41597-025-04689-3.

[10] S. S. Stevens, J. Volkmann, E. B. NewMan: *A scale for the measurement of the psychological magnitude pitch*, American. Journal of Psychology 8.3 (1937), pp. 185–190, doi: 10.1121/1.1915893.

[11] S. Wang, A. Sekey, A. Gersho: *An objective measure for predicting subjective quality of speech coders*, IEEE Journal on Selected Areas in Communications 10.5 (1992), pp. 819–829, doi: 10.1109/49.138987.

[12] E. Zwicker: *Subdivision of the audible frequency range into critical bands (frequenzgruppen)*, Journal of the Acoustical Society of America 33.2 (1961), p. 248, doi: 10.1121/1.1908630.