Improving PPG-based Heart-Rate Monitoring with Synthetically Generated Data

(Article begins on next page)

Improving PPG-based Heart-Rate Monitoring with Synthetically Generated Data

Alessio Burrello*, Daniele Jahier Pagliari†, Marzia Bianco*, Enrico Macii§, Luca Benini*‡, Massimo Poncino†, Simone Benatti¶
* DEI, Università di Bologna, Bologna, Italy
† Department of Control and Computer Engineering, Politecnico di Torino, Turin, Italy
‡ Integrated Systems Laboratory, ETH Zurich, Switzerland
§ Interuniversity Department of Regional and Urban Studies and Planning, Politecnico di Torino, Turin, Italy
¶ Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Italy
Emails: name.surname@unibo.it, name.surname@polito.it

***Abstract*—Improving the quality of heart-rate monitoring is the basis for a full-time assessment of people's daily care. Recent state-of-the-art heart-rate monitoring algorithms exploit PPG and inertial data to efficiently estimate subjects' beats-per-minute (BPM) directly on wearable devices. Despite the easy-recording of these signals (e.g., through commercial smartwatches), which makes this approach appealing, new challenges are arising. The first problem is fitting these algorithms into low-power memory-constrained MCUs. Further, the PPG signal usually has a low signal-to-noise ratio due to the presence of motion artifacts (MAs) arising from movements of subjects' arms. In this work, we propose using synthetically generated data to improve the accuracy of PPG-based heart-rate tracking using deep neural networks without increasing the algorithm's complexity. Using the TEMPONet network as baseline, we show that the HR tracking Mean Absolute Error (MAE) can be reduced from 5.28 to 4.86 BPM on PPGDalia dataset. Noteworthy, to do so, we only increase the training time, keeping the inference step unchanged. Consequently, the new and more accurate network can still fit the small memory of the GAP8 MCU, occupying 429 KB when quantized to 8bits.**

I. Introduction and Related Works

Wearable devices for health monitoring are gaining remarkable traction in the medical IoT era [1], by virtue of integrated sensors, computationally efficient platforms, and more advanced algorithms. Hearth Rate (HR) is one of the most important vital signs monitored by clinicians since it is involved in many diagnostic scenarios. Alongside punctual HR measures, which can be performed at home or in ambulatory, long-lasting heartbeat monitoring can prevent potentially serious diseases and keep critical patients' health conditions in check. First-generation devices for continuous HR monitoring were based on 3 leads ECG measure, mounted on a bulky and uncomfortable chest band. This approach is still used for 24 hours Holter monitors, but it is not suitable for a daily life assessment.

More recently, the use of wrist-worn Photoplethysmographic (PPG) sensors has been increasing, pushed mainly by the development of low-cost fitness trackers [2]. PPG relies on one or more photodiodes measuring the light emitted by pulsating LEDs and absorbed/reflected by blood vessels. While these sensors provide an advantage in terms of wearing comfort, their accuracy is negatively affected by Motion Artifacts (MA), i.e., sensor pressure variability and external light leaks caused by arms' movements. Furthermore, especially during intense workouts, these effects are superposed to a higher blood flow variability, complicating the HR measure. To cope with MA, several solutions have been presented in the literature, leveraging adaptive filtering and/or sensor fusion techniques [3], [4].

The seminal work of [3] paved the way in this field, open-sourcing the first dataset (IEEE Training), and introducing an algorithm based on signal decomposition and filtering, TROIKA, achieving a MAE of 2.34 BPM on the new dataset. In 2015, [4] improved TROIKA using spectral difference with the acceleration to mitigate the effect of Motion Artifacts (MAs), reducing the MAE to 1.28 BPM. In 2017 and 2018, two works [5], [6] introduced the Wiener filtering to clean PPG signals, further cutting the MAE to 0.99 BPM. More recently, [7] presented a time-domain algorithm which achieves 4.6 BPM of MAE on the larger and more complex PPGDalia dataset (open-sourced in [8]) with a 5-steps pipeline of light linear transformations.

Noteworthy, all these approaches give good results but lack generality, since they rely on a heavy manual tuning of the hyper-parameters, and their performance often does not generalize on datasets different to those on which they are tuned, as demonstrated in [8]. Moreover, many of these methods are too complex for real-time implementations on wearable devices.

A promising alternative consists in resorting to Deep Learning (DL) techniques [9], [8], [10], which are more robust than conventional approaches, since they can capture relations in the data difficult to express in closed form, and are rapidly becoming state-of-the-art also in this field. The works of [10], [11], [8] started this new trend in 2019, achieving results comparable to classical methods with Convolutional Neural Networks (CNNs) applied to frequency data or CNN+LSTM (Long-Short Term Memory) to time-domain data. Recently, [12], [9] explored the automatic optimization of DL models for HR monitoring through Neural Architecture Search (NAS), achieving state-of-the-art performance on the PPGDalia dataset.

One of the main factors preventing DL methods from achieving even further performance improvements on this task is the scarcity of training data. In fact, the accuracy of deep neural networks usually keeps improving with more data, well beyond the point where conventional algorithms would saturate. However, collecting "real-world" PPG-based HR monitoring data is a costly procedure, mainly due to the need to measure

This work was supported in part by the European H2020 FET Project OPRECOMP under Grant 732631.

the ground truth HR with a medical-grade ECG device. In this work, we tackle the problem with *data augmentation* techniques, which are popular in other domains [13], [14], but have never been used for this task. Our main contributions are:

1) We propose 2 novel methods to generate synthetic PPG and acceleration data, which extend the training golden HR labels range and improve the accuracy on subjects with extremely high/low average BPM.
2) We benchmark 4 additional data augmentation techniques from the time-series literature to increase the size of a training dataset for PPG-based HR detection, assessing their effect on the state-of-the-art TEMPONet architecture, a popular DL model for biosignal processing tasks.
3) We demonstrate that data augmentation can reduce the Mean Absolute Error of the model by up to 0.42 BPM (7.95%) on TEMPONet, while maintaining the inference cost *identical*.

II. Materials & Methods

While there exist a significant literature on optimized algorithms for PPG-based HR monitoring (see Section I), to our knowledge, no one explored the orthogonal direction of *data-driven optimizations* for this task, i.e., improving the quality and increasing the amount of data fed to DL models, without modifying their structure.

Data Augmentation (DA) is a popular data-driven technique that generates modified copies of the original data to increase the training set size. Namely, new data $\bar{x}_i$ are generated from original ones x_i by adding noise or by means of scale, shape, or time-axis transformations. Typically, DA is employed to reduce overfitting and improve the generalization of machine learning and deep learning models.

In this work, we apply for the first time four classical DA schemes for time-series, and two new tailored transformations, to the PPG and 3D-acceleration data processed by a DL HR monitoring algorithm. Fig. 1 shows an example of the 6 augmentations considered, described in detail in the following.

A. Classical Data Augmentation

We selected four popular DAs for time-series which have shown remarkable results in improving the accuracy of time-series processing tasks such as activity recognition, traffic estimation, etc [13], [14]. Fig. 1 reports the synthetic data generated through these algorithms in panels 1-4.

Jittering consists in sampling a value from a Gaussian distribution with $\mu = 0$ for each time sample t and adding it to the raw signal. The augmented signal is a noisy version of the input one, i.e.:

$$X_t = X_t + \mathcal{N}(0, \sigma) \qquad \forall t$$

Scaling samples a single value from a Gaussian distribution with $\mu = 1$, and multiplies the entire input window by it. The resulting signal is identical to the original one but with enlarged/reduced amplitude:

$$X = X \times \mathcal{N}(1, \sigma)$$

Magnitude Warping, similar to Scaling, alters the magnitude of the input window but does so by multiplying the original signal envelope with a cubic spline that interpolates the points randomly generated from a Gaussian distribution with $\mu = 1$:

$$X = X \times CubicSpline(0...T, \mathcal{N}(1, \sigma))$$

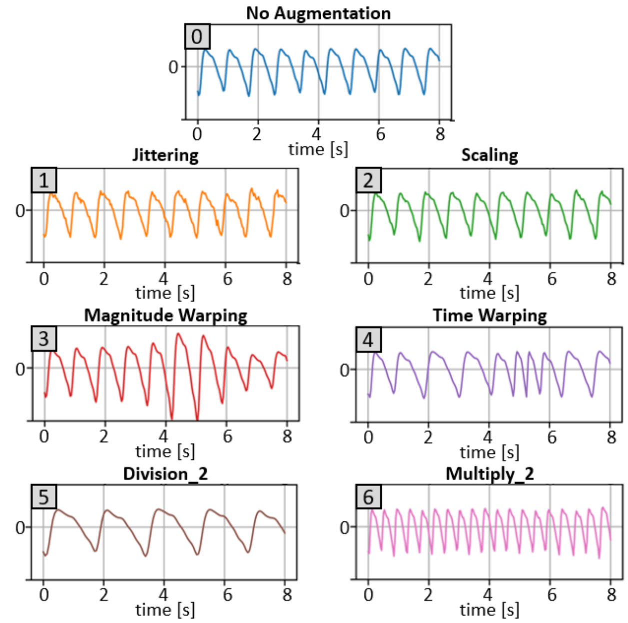

Fig. 1. Resulting synthetic signals from the application of the six different data augmentation algorithms to a 8 seconds window of the PPG-Dalia dataset.

Lastly, **Time Warping**, instead of changing the magnitude of the signal, alters it on the time axis. Specifically, the time distance between two consecutive samples is changed from the original constant ($1/f_s$, where f_s is the sampling frequency) to $CS_t - CS_{t-1}$, where CS_t are the points of a randomly generated cubic spline. As shown in Fig. 1, the resulting augmented signal is stretched in some regions and compressed in others:

$$X = Interp(\,Cumulative(\,CubicSpline(0...T, \mathcal{N}(1, \sigma))), X)$$

All signals have been normalized between -1 and +1 before applying the transformations. The HR label of each window has been kept unchanged. We employed a standard deviation $\sigma = 0.05$ for Jittering and Scaling, and $\sigma = 0.5$ for Time/Magnitude Warping. We used these default values since this work aims to preliminary assess the effectiveness of data augmentation for HR monitoring. The systematic exploration of transformation hyper-parameters through cross-validation will be the subject of future work.

B. Task-Specific Data Augmentation

From our experience on training DNNs for HR monitoring, we noticed that subjects at the two boundaries of the BPM range, i.e., with very low or very high average BPM, were often badly tracked, given that most of the training data corresponded to less extreme values. To cope with this problem, we propose two novel task-specific data augmentation algorithms that not only aid the reduction of overfitting, but also widen the BPM range in the training set.

With the **Divide_2** augmentation, we extend the lower bound of the BPM range by randomly selecting, for each input window of length T, a continuous section of length $T/2$, and stretching it in time back to the original window size (0 to T) with resampling. The corresponding BPM label is also halved.

On the other hand, we use the **Multiply_N** augmentation to extend the upper bound of the BPM range. In this case, we consider multiple upscaling factors (1.1× to 2×), since HR prediction with high BPMs is more difficult per se (due to more

TABLE I
AUGMENTATION CONFIGURATIONS EXPLORED IN THIS WORK.

Name	Aug. [×]	Aug. Type	Perc. [%]
Division	2×	Divide_2	100%
Multiplication	3×	Multiply_[1.4-2.0, 0.2]	200%
Div_Mul_1	4×	Divide_2	100%
		Multiply_[1.4-2.0, 0.6]	200%
Standard_1	6×	Jittering	150%
		Scaling	150%
		Time Warping	100%
		Magn. Warping	100%
Standard_2	9×	Jittering	200%
		Scaling	200%
		Time Warping	200%
		Magn. Warping	200%
All Augs	10×	Jittering	200%
		Scaling	200%
		Time Warping	100%
		Magn. Warping	100%
		Divide_2	100%
		Multiply_[1.4-2.0, 0.6]	200%
Div_Mul_2	11×	Divide_2	100%
		Multiply_[1.2-2.0, 0.1]	900%

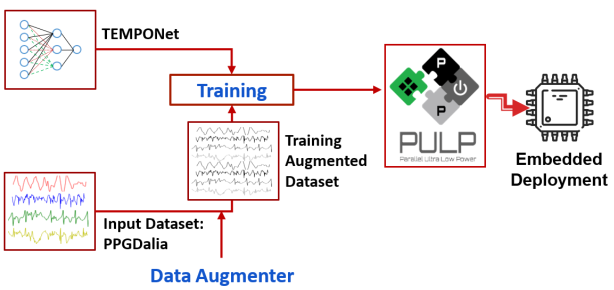

Fig. 2. Flow of the proposed work.

MAs). To do so, we create a temporary twin dataset composed of windows of length $2T$, obtained from the concatenation of two consecutive original windows, and having as ground-truth BPM label the one computed on the entire $2T$ period. Then, based on the desired multiplication factor, we sample a randomly selected portion of the window of length $1.1T$-$2T$. Lastly, these data are re-projected on the original length-T time window with linear interpolation, and the BPM label is upscaled accordingly.

C. Training Setup

Starting from the transformations described above, we explore seven DA configurations, to span a wide range of training dataset sizes (from 2× to 11× of the original one). Table I details the configurations, reporting the final training set size increase compared to the not-augmented one, the type of DA(s) applied, and the percentage of new samples generated with each transformation, relative to the original size. Multiply_[x-y, z] indicates the minimum (x), maximum (y), and step (z) of the considered multiplication factors. For instance, Multiply_[1.4-2.0,0.2], 200% refers to an augmentation of 200% of the original dataset with synthetic data with HR frequency multiplied by 1.4, 1.6, 1.8, and 2.0 (50% new samples for each different factor).

The overall training and deployment flow followed in our work is shown in Fig. 2. Compared to more classical flows where the architecture of a neural network is optimized (manually or automatically) for the target task [8], [9], [15]), we act solely on the input data, keeping the model unchanged. In particular, we employ the PPG-Dalia dataset [8] to train and test our approach, and we consider the TEMPONet network as

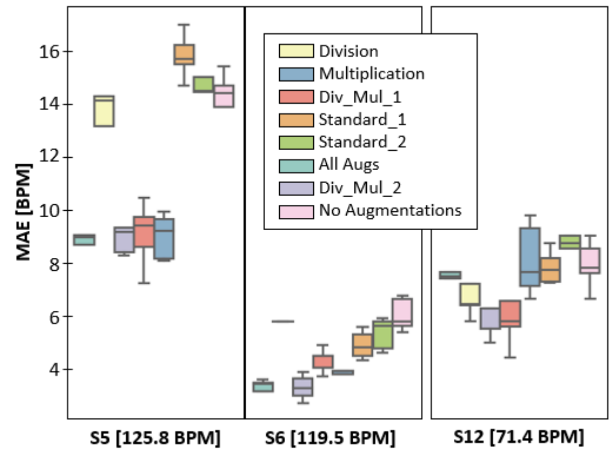

Fig. 3. Average MAE of all augmentation configurations and of not-augmented training on the four subjects with most extreme average BPM (in caption).

HR tracking model [16], [9]. Our main addition to the flow is the *Data Augmenter* block, which takes as input the original training dataset, the transformations to apply, and a target size, and produces a new training dataset, with both original and synthetically generated data. This step is *orthogonal* to any topology transformation (e.g., the architecture optimization of [9], [15]). To stress this fundamental point, the trained TEMPONet is deployed on the GAP8 platform [17] to show that its inference-time complexity remains unchanged compared to a not-augmented training, thus remaining compatible with real-time edge execution.

To compare our approach with state-of-the-art algorithms, we use the same leave-one-subject-out train/test split procedure proposed in the original PPGDalia paper [8], repeating each training 10 times to increase the robustness of the results. We perform 500 training epochs, with an early stop after 20 epochs without any improvements in the validation loss. We use the Adam optimizer (learning rate = 1e-3, weight decay = 0), and a batch size of 128. We minimize the $\log\cosh$ loss to target the mean absolute error (MAE) as an accuracy metric.

III. EXPERIMENTAL RESULTS

We evaluate our flow on the largest public PPG-based HR tracking dataset, PPGDalia [8]. The dataset comprises 37.5 hours of data recorded from 15 subjects. The inputs are PPG-sensor and 3D-accelerometer data, associated with golden HR values collected through an ECG chest band. All experiments are performed using Python 3.6 and PyTorch 1.6. The target GAP8 platform has 64 kB of Tightly-Coupled Data Memory (TCDM), 512 kB of RAM, and 8 MB of external memory. It features 8 computing cores and 1 controller core, all RISC-V based, with specialized instructions for efficient DNN inference [17].

A. Improvements on Extremely High and Low BPM Ranges

First, we observe the effects of our DA on subjects with extremely high or low BPMs, badly represented in the training data. Namely, we select the 3 subjects whose average HR differs more ($>$ 20 BPM) from the corresponding training set (i.e., the other 14 subjects). Those are the two subjects with the largest average HR overall, i.e, S5 (125.8 BPM) and S6 (119.5 BPM), and the one with the lowest average, i.e., S12 (71.4 BPM). Fig. 3 shows the results of 10 training runs with the seven proposed augmentation configurations and with the not-

TABLE II
COMPARISON WITH STATE-OF-THE-ART DEEP LEARNING PPG-BASED HR MONITORING ALGORITHMS.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	Mean
DeepPPG [8]	7.73	6.74	4.03	5.90	18.51	12.88	3.91	10.87	8.79	4.03	9.22	9.35	4.29	4.37	4.17	7.65
NAS-PPG [12]	5.46	5.01	3.74	6.48	12.68	10.52	3.31	8.07	7.91	3.29	7.05	6.76	3.84	4.85	3.57	6.02
TEMPONet	4.37	3.74	2.43	5.49	13.48	5.71	2.23	7.86	8.94	3.32	5.34	7.71	2.03	2.94	3.58	5.28
All Augs	4.97	4.34	2.39	6.14	9.41	3.63	2.23	9.14	10.98	3.40	5.27	7.64	2.05	2.84	3.61	5.20
AugTEMPONet	4.37	3.74	2.43	5.49	9.41	3.63	2.23	7.86	8.94	3.32	5.34	7.64	2.03	2.94	3.58	4.86

augmented dataset. Notably, all DAs involving the expansion of the upper limit of the training BPM range (i.e., Multiplication, Div_Mul_1, Dim_Mul_2, and All Augs) reduce the MAE of S5 and S6. The subject that most benefits from the injection of synthetic data is S5, i.e., the one with the highest average HR in the dataset. On this subject, using a training dataset 3$\times$ larger than the original one, with synthetic HR speedups of 1.4x and 2.0x, reduces the MAE from 13.48 to 8.82. Overall, the best augmentation scheme (Div_Mul_2), which expands both the upper and lower bounds of the training BPM range, reduces the MAEs of all three subjects.

B. State-of-the-art comparison with DNNs

This section analyzes the overall performance of applying data augmentation to TEMPONet. Table II reports the per-subject performance of three state-of-the-art networks, TEMPONet, DeepPPG [8], and NAS-PPG [12]. In the lower part of the table, we report the results obtained applying all DA transformations (All Augs) to all subjects, and with a simple subject-selective DA policy. Specifically, the latter augments the training set only for the three "extreme" subjects S5, S6, and S12, following the same rationale explained above.

The results show that it is possible to improve the TEMPONet performance without changing the network topology. Using the *All Augs* data augmentation, we slightly reduce the MAE from 5.28 BPM to 5.20 BPM. Specifically, we enormously improve subjects S5 and S6, whereas we have slightly higher MAE on subjects such as S8 and S9. This shows that, for subjects with a HR range already well-represented in the original data, DA is not always beneficial, since it can increase the noise in the training data and hinder the learning of a relatively limited capacity model such as TEMPONet (only 423k parameters). To counter this detrimental behavior, we decided to test the aforementioned subject-selective policy, which in fact reduces the overall MAE by 0.42 BPM. Even more important than the overall MAE reduction is the fact that the proposed DAs lead to improvements on otherwise badly-tracked subjects. In a real scenario, this is a great advantage, given that a new unseen subject may show BPM values outside the range available in the training dataset. Most of this improvement comes from the newly proposed task-specific transformations. In fact, the Div_Mul_2 augmentation alone reduces the error on S5, S6, and S12 from 13.48, 5.70, and 7.71 BPM to 9.26, 3.69, and 7.20 BPM, respectively, with an average reduction of 2.25 BPM.

Notably, this analysis is orthogonal to the application of additional post-processing steps [9], or NAS [15], which could still reduce the MAE, but are outside the scope of this work.

C. Edge Deployment Analysis

Table III reports deployment and training-related metrics of state-of-the-art networks as well as of the baseline and augmented TEMPONet network. First, we observe that TEMPONet already outperforms other DNN-based algorithms, reaching a lower MAE (5.28 BPM vs 6.02 BPM) with fewer parameters and MACs. Unfortunately, the original papers did not report the state-of-the-art network's energy, latency, and training time. On the other hand, the table underlines the crucial point of this work. Our new AugTEMPONet reaches a better MAE (4.86 vs. 5.28 MAE) than its not-augmented counterpart while keeping inference latency and energy constant. Given the more extensive training dataset, the training time per epoch increases from 13.08s to 114.7s. However, training is a one-time, offline operation for this task, therefore not impairing the battery life of the edge device on which the algorithm is deployed.

TABLE III
DEPLOYMENT-RELATED METRICS OF ANALYZED NETWORKS. TRAINING TIME IS MEASURED ON A GEFORCE GTC 1080TI 12GBS AND REFERS TO ONE EPOCH. LATENCY AND ENERGY ARE MEASURED ON GAP8.

Model	Par.	MACs	Train T.	Inf. E.	Inf. Lat.
DeepPPG	60 M	480 M	-	-	-
NAS-PPG	800k	16.3 M	-	-	-
TEMPONet	423k	13.5 M	13.08 s	1.2 mJ	23.2 ms
AugTEMPONet	423k	13.5 M	114.7 s	1.2 mJ	23.2 ms

IV. CONCLUSIONS

Executing accurate HR monitoring algorithms at the edge allows continual personalized health monitoring. To keep the model complexity low and increase its accuracy, we have proposed two new augmentations tailored to HR that, combined with classical transformations, improve the performance of state-of-the-art DNNs for HR without increasing their energy consumption. Specifically, the MAE of our baseline network TEMPONet can be reduced from 5.28 to 4.86 BPM while still consuming only 1.2 mJ per inference on the GAP8 MCU.

REFERENCES

[1] N. Sviridova *et al.*, "Human photoplethysmogram: new insight into chaotic characteristics," *Chaos, Solitons & Fractals*, vol. 77, pp. 53–63, 2015.

[2] Fitbit. Fitbit charge 4. [Online]. Available: https://www.fitbit.com/global/us/products/trackers/charge4

[3] Z. Zhang *et al.*, "Troika: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise," *IEEE TBME*, 2014.

[4] Z. Zhang, "Photoplethysmography-based heart rate monitoring in physical activities via joint sparse spectrum reconstruction," *IEEE TBME*, 2015.

[5] A. Temko, "Accurate heart rate monitoring during physical exercises using ppg," *IEEE TBME*, 2017.

[6] H. Chung *et al.*, "Finite state machine framework for instantaneous heart rate validation using wearable photoplethysmography during intensive exercise," *IEEE JBHI*, 2018.

[7] N. Huang *et al.*, "Robust ppg-based ambulatory heart rate tracking algorithm," in *2020 42nd EMBC*, 2020.

[8] A. Reiss *et al.*, "Deep ppg: large-scale heart rate estimation with convolutional neural networks," *Sensors*, vol. 19, no. 14, p. 3079, 2019.

[9] M. Risso *et al.*, "Robust and energy-efficient ppg-based heart-rate monitoring," in *2021 IEEE ISCAS*, 2021.

[10] D. Biswas *et al.*, "Cornet: Deep learning framework for ppg-based heart rate estimation and biometric identification in ambulant environment," *IEEE TBIOCAS*, 2019.

[11] X. Chang *et al.*, "Deepheart: Accurate heart rate estimation from ppg signals based on deep learning," in *2019 IEEE MASS*, 2019.

[12] S. B. Song *et al.*, "Nas-ppg: Ppg based heart rate estimation using neural architecture search," *IEEE Sensors Journal*, 2021.

[13] K. M. Rashid *et al.*, "Times-series data augmentation and deep learning for construction equipment activity recognition," *Advanced Engineering Informatics*, vol. 42, p. 100944, 2019.

[14] Q. Wen *et al.*, "Time series data augmentation for deep learning: A survey," *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021.

[15] A. Burrello *et al.*, "Q-ppg: Energy-efficient ppg-based heart rate monitoring on wearable devices," *IEEE TBIOCAS*, 2021.

[16] M. Zanghieri *et al.*, "Robust real-time embedded emg recognition framework using temporal convolutional networks on a multicore iot processor," *IEEE Transactions on Biomedical Circuits and Systems*, 2019.

[17] E. Flamand *et al.*, "Gap-8: A risc-v soc for ai at the edge of the iot," in *2018 IEEE 29th ASAP*. IEEE, 2018.