On-device subject recognition in UWB-radar data with Tiny Machine Learning

Massimo Pavan[1], Armando Caltabiano[2], and Manuel Roveri[1]

[1] Politecnico di Milano, Milano, IT {massimo.pavan,manuel.roveri}@polimi.it
[2] Truesense s.r.l., Milano, IT armando.caltabiano@truesense.it

Abstract. Tiny Machine Learning (TinyML) is a novel research area aiming at designing machine and deep learning models and algorithms able to be executed on tiny devices such as Internet-of-Things units, edge devices or embedded systems. Smart pervasive devices are rapidly becoming omnipresent in our every-day life, and TinyML and its paradigm of executing everything on-device (and thus not moving the data from where they are collected) has been crucial in designing algorithms and applications that enhance the privacy of users.
From this perspective, radar sensors are currently emerging as a valid alternative to common sensors (e.g. microphones, cameras...). Given the impossibility to recognize precisely the identity of the user, they can be used in cases where it is important to recognize the presence or the behaviour of human beings while guaranteeing at the same time to preserve their privacy. UltrawideBand (UWB), in particular, is a radar technology that is particularly promising for use in pervasive systems. Indeed, its precision, low energy consumption and fastness are particularly suitable for privacy-preserving embedded applications.
We introduce here, for the first time in the literature, a TinyML solution integrating pre-processing and tiny convolutional neural network for subject recognition (i.e., recognizing the age-class of the target) through the analysis of UWB-radar data.
The proposed solution has been successfully tested on a real-world application of in-car subject recognition.

Keywords: TinyML · UWB-radar · Deep learning

1 Introduction

In recent years the diffusion of tiny devices, such as Internet-of-Things (IoT) units, edge devices and embedded systems, representing the technological asset of the "computing everywhere" paradigm [1][2], have been constantly rising. From this perspective, the scientific trend is to move the processing (and in particular the intelligent processing) as close as possible to where data are generated to increase the autonomy of tiny devices, reduce the latency and the required transmission bandwidth they require, while increasing the energy efficiency [3][4]. The new Machine and Deep Learning solutions (MDL) able to be executed on these tiny devices must take into account the severe constraints on memory (the

available RAM is in the order of the MB), computation (the MCU frequency is in the order of the MHz), and power consumption (typically < 0.1 W) of these devices.

The role of Tiny Machine Learning (TinyML) is to design, develop and deploy MDL models and algorithms for tiny devices. TinyML solutions present in the literature typically introduce tiny MDL architectures and approximate-computing solutions (such as quantization [5], pruning[6], and early-exit mechanisms[7][8]) to fit the severe technical constraints characterizing these tiny devices.

The aim of this paper is to introduce a TinyML solution for subject recognition (i.e., recognizing the age-class of a person) on UWB-radar data. The proposed solution, which extends what introduced in [9] for person detection in UWB-radar data, relies on a preprocessing phase to highlight relevant features and on a suitably-defined tiny convolutional neural networks based on tiny dilated convolutional blocks and quantization of the CNN architecture to reduce the computational and memory demands (of both weights and activations).

The proposed solution has been successfully tested on a real-world in-car subject recognition application. In particular, the proposed UWB-based TinyML solution for the in-car presence-detection has been successfully deployed and tested in real-world conditions on an ESP32 microcontroller unit (4 MB of flash memory, 512 KB of S-RAM memory), equipped with a UWB-radar module comprising only one pair of antennas.

The paper is organized as follows. Section 2 describes the related literature, while Section 3 introduces the proposed TinyML solution for UWB-based subject recognition. Section 5 details the problem definitions and the experimental results for the in-car presence detection scenario. Finally, Section 6 draws the conclusion and describes the future research directions in this field.

2 Related literature

This section describes the related literature in the field of TinyML (Section 2.1) and the available UWB-radar solutions for presence detection and activity recognition (Section 2.2). Given the novelty of the proposed problem, no solution is present in the literature for subject recognition. We emphasize that, the only TinyML solution able to process UWB-radar data available in the literature is our previous work on presence detection[9].

2.1 TinyML

The research in the field of machine learning for embedded systems and IoT units is mainly addressed from two different point of view: the development of custom hardware and the design of approximated MDL solutions. We here concentrate on approximated MDL solutions.

The design of *approximated machine/deep learning solutions* capable of addressing the strict technological constraints of embedded and IoT units is a relevant and continuously-growing research field. The techniques introduced in

this area can generally fall within the field of TinyML [10][11]. Most of the related literature focuses on the approximation of Convolutional Neural Network algorithms. For example, in [12] a methodology to explore sparse CNN architectures that could be executed on Microcontroller units (MCUs) was introduced, whereas [13] proposed Bonsai, a decision tree-based technique to perform CNN-inference efficiently on Arduino boards. In addition, pruning of channels and layers of CNNs has proven to be a successful [14][15] in reducing the memory and computational demand.

A different approach to approximate CNNs is to reduce the memory required by the solution through the use of quantization, which exploits limited-precision data types [16][17] for the CNN weights and, possibly activations. In such a direction, [18] combined both task dropping and precision scaling techniques to design approximated CNNs able to be executed in IoT units.

Other solutions focus on reducing the mean inference time of deep neural networks. Adaptive Early Exit [8] and Gate-Classification CNNs [7] are an example of such solutions.

2.2 UWB-radar usage

The literature about the usage of artificial intelligence with UWB-radar data mainly concentrates on tasks similar to person detection and human activity recognition (HAR). UWB-radar were used also for human sensing and vital parameter estimation. In this related literature, we focus our attention on UWB-radar solutions that rely on a single receiving antenna.

Presence detection Most of the solutions for presence detection based on uwb-radar relies on thresholds or statistical approaches to distinguish between empty records and records where a human is present [19][20]. These solutions are usually heavily dependant on the dataset, and thus fails to generalize. Finally, anti-abandon systems for cars based on radar can be also found in the literature [21][22] but they do not rely on uwb-radar and they are only meant to detect the presence of a general subject in a car.

Human Sensing Of particular interest is the work of [23], who has used UWB radar data for wireless human sensing and personal identification. In this field, the possibility to recognize the breath and heartbeats of the targets is proven also in [24] and [25]. Nevertheless, in each of these researches the target is standing in front of the radar without the possibility of moving, making it very difficult to generalize in a general use-case scenario.

3 Problem formulation and motivation

3.1 Acquiring and processing UWB-radar data

Let $S \in \mathbb{R}^{N \times M}$, with $M, N \in \mathbb{N}$, be the output of the UWB-radar receiving-antenna installed on the device, being N the collected number of radar scans and M the number of "bins" characterizing the acquisitions of the antenna. In more

detail, the value $S[i, j]$ with $i = \{1, \ldots, N\}$ and $j = \{1, \ldots, M\}$ represents the energy acquired by the i-th scan at the j-th bin. We emphasize that $N = W \cdot f_r$, being f_r the UWB-radar frame rate (i.e., the number of acquisitions per seconds) and W the acquisition time horizon (in seconds), while M represents the number of "quantized" distances in the acquisition range, i.e., from MIN_RANGE to MAX_RANGE, of the UWB-radar antenna[3]. An example of the acquisition of S is shown in Figure 1.

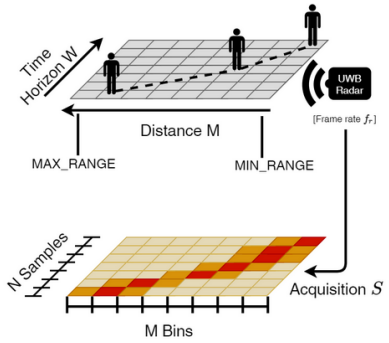

Fig. 1. The acquisition of matrix S by the UWB-radar antenna.[9]

3.2 Dataset collection

For this work, the dataset presented in [9] was extended with new data. In all the recordings the device was deployed above one of the back lateral windows of the car (the radar can detect subjects in a $\pm 60°$ cone from where it's directed). The dataset contains records with 0, 1, 2 or all 3 seats occupied by a target. The total amount of acquired samples is 429, divided into 163 records with a child present in the first seat, 220 records with the first seat empty, and 46 records with an adult in the first seat. Figure 2 describes the positioning and the cone of view of the device during the data-collection phase.

3.3 Recognizing subjects through UWB-radar data

A high number of studies have enlighted the possibility to reliably estimate the breathing frequencies of human targets with the use of UWB-radar data [25, 23].

The standard breathing frequencies of each age category at rest have been estimated as illustrated in the table 1, where the data have been taken from [26].

[3] Each value represents the amount of energy of the reflected radar wave. MIN_RANGE, MAX_RANGE and M are parameters depending on the specific radar device used and on its configuration.

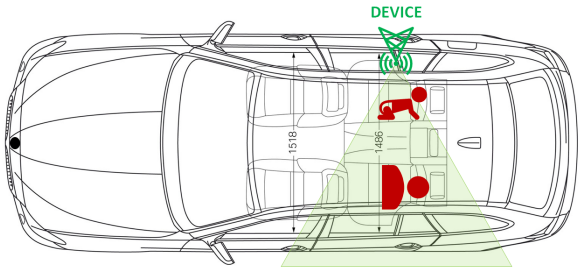

Fig. 2. The acquisition campaign for this experimental analysis[9]

Class	Avg. respiratory rate at rest
birth to 6 weeks	30-40 breaths per minute
6 months	25-40 breaths per minute
3 years	20-30 breaths per minute
Adults	15-18 breaths per minute
Elderly $\geq$ 65 years old	12-28 breaths per minute

Table 1. Average respiratory rates

From this perspective, the breath frequencies of the targets from UWB-radar data could be used to to distinguish among different age-classes (to recognize subjects). Since the radar module makes scans at a frequency of 5Hz, it is possible to match each frequency bin of the Fast Fourier Transform (FFT) data with a frequency range, calculated as:

$$\text{dimension of each frequency bin} = \frac{5\text{Hz}}{128} = 0.039 \approx 0.04$$

For example, the Hz range of the first bins will be 0 - 0.04 Hz, while the last one will be 4.96- 5 Hz

From these estimations, the table 1 have been updated to include the bins where it is possible to expect the breathing frequencies to be recognizable.

Class	Avg. respiratory rate at rest	Hz	Expected bins
Children	20-40 breaths per minute	1/3 - 2/3 Hz	8,33 - 16, 66
Adults	15-18 breaths per minute	1/4 - 0.3 Hz	6,25 - 8

Table 2. Average respiratory rates and expected bins

In order to check these hypotheses, for each class the FFT data of some scans belonging to the dataset used for the experiment in section 5 have been visualized as a heatmap, searching for peaks in energy around the expected bins.

The following visualizations represent the euclidean norm of the real and imaginary parts of the FFT data of specific scans, in which only one target is present. The first bins have been artificially set to 0 in order to better display the interesting portion of data.

Note that in order to better display the peaks, in the visualizations the scale is not fixed.

In Figure 3 the data of two scans containing only one Adult in the first seat are reported. Around bins 6-8, in which the breathing frequency of the target should reside, it is possible to observe some peaks in the data, but overall, especially in the second record, they are not easily distinguishable.

The visualizations have been repeated also for children (Figure 4). In these visualizations it is much more difficult to clearly distinguish peaks traceable to the breathing frequencies of toddlers and babies (expected bins 8-17). It's also interesting to note that in that visualization the scale is almost an order of magnitude smaller with respect to the adults' records: this could mean that the breathing is not absent, but that it is much more difficult to distinguish among records of people with different age-class it from the noise of the recording. Nevertheless, the magnitude of the signal could in principle be a relevant aspect for the classification of the record. Anyway, even in the same class of records, there are significant differences between one record and another. There is no clear unique behaviour for the same type of data, or at least is hardly recognizable by watching the graphs.

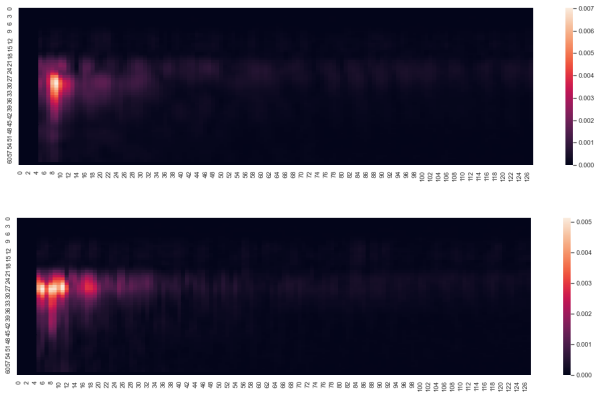

Fig. 3. Euclidean norm of FFT of two adult-0-0 scans.

4 The proposed TinyML solution for UWB-radar based subject recognition

The proposed TinyML solution for subject recognition based on UWB-radar comprises two main modules: a pre-processing module and and tiny deep convolutional neural network called TyCNN-C. These two modules, which are detailed in the sequel, have been jointly designed and developed to maximize the recognition accuracy, while satisfying the strict technological constraints of tiny devices.

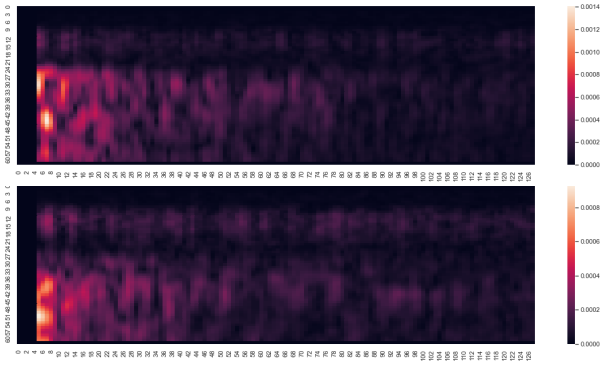

Fig. 4. Euclidean norm of FFT of two toddler-0-0 scans.

4.1 Pre-processing

The pre-processing module comprises the following three steps: Fast-Fourier Transform, low-pass frequency selection and data normalization. This module has not been substantially modified from the TyCNN for presence detection implementation, and thus we remand to [9] for the detailed description.

The first step of pre-processing aims at computing the FFT S_f of S. The considered FFT algorithm is the Cooley-Tukey algorithm [27]. The FFT is computed on all the M rows of S.

The second step aims at selecting only a sub-range of the frequencies in S_f to reduce the memory and computational demands of the next tiny convolutional neural network (see Section 4.2). Being $f_l < f_r/2$ the selected cut-off frequency, the goal of this module is to remove from S_f the rows corresponding to the frequencies larger than f_l. In this application setting, assuming the relevance of the breathing frequencies discussed in section 3.3, f_l can be scaled down (with a negligible loss of accuracy) to roughly 2Hz.

Finally, the last pre-processing step comprise a log-scale transformation and a Z-score normalization (where mean and standard deviation are computed on the training set of the TyCNN-C).

4.2 TyCNN-C: deep convolutional neural network

We introduce here TyCNN-C, which is the TinyML CNNs for subject recognition with UWB-radar data. The proposed solution, which extends the TyCNN design proposed in [9] for person detection, has been carefully designed to satisfy the technological constraints on the memory footprint $\bar{m}$ (in KB) and computation $\bar{c}$ (number of operations) of the target tiny devices.

An overview of the TyCNN-C is shown in Figure 5, while the TyCNN-C architecture and the quantization mechanism are summarized in what follows.

Being X_I the input of the TyCNN-C, the processing layers can be summarized as follows:

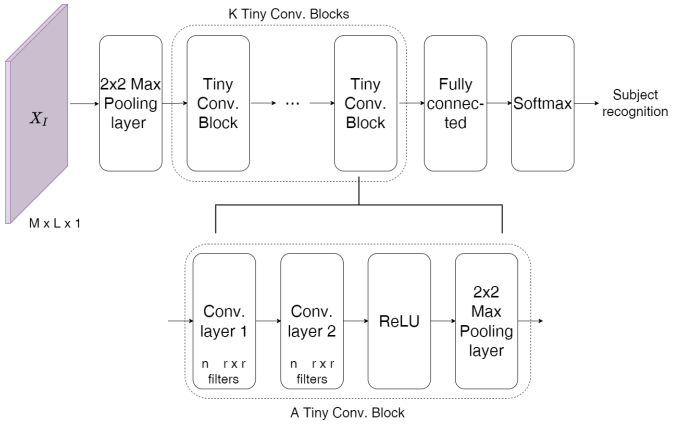

Fig. 5. The general architecture of the tiny convolutional neural networks TyCNN-C.

A 2×2 Max Pooling layer this layer aims at reducing the size of the input X_I. In more detail, the goal of this layer is to reduce the memory demand of intermediate activations as well as the number of operations required by the TyCNN-Cs to compute the inference.

A sequence of K Tiny Convolutional Blocks the Tiny Convolutional Blocks (TCBs) introduced for the TyCNN, were reused for the TyCNN-Cs architecture. Each block comprises the four following steps:

- two convolutional layers comprising n square $r \times r$ dilated filters with dilation rate equal to 2;
- the ReLu activation function;
- a 2×2 Max Pooling layer.

In the considered subject-recognition application described in Section 5, the following configuration of the TBCs have been considered: $K = 2$, $n = 14$ and $r = 7$.

A fully-connected layer The aim of this last layer is to provide the final classification of the TyCNN-C. In more detail, this layer is composed of a flattening layer, a dropout layer (with dropout rate equals to 0.3), and a single dense layer. Differently from the previous TyCNN design, the dense layer is characterized by softmax activation.

For the training we considered the Categorical Crossentropy as loss function, while Adam was selected as optimizer. The learning rate was set to 0.3e-4, while the number of training epochs was set to 400. Given the fact that the distribution of the classes in the training set were really skewed, the errors on each record were weighted, using weights inversely proportional to the representation of the corresponding class in the dataset. Once the TyCNN-C has been trained, the full-integer post-training weight quantization algorithm introduced in [28] has been used to transform the 32-bit floating-point weights into 8-bit integers. The same quantization scheme has been also applied to inputs and activations.

Table 3. The detailed memory footprint (with an 8-bit data type) and the number of operations of the TyCNN-C for the in-car scenario. To optimize the memory, two arrays only are used to store the activations (an @ marks the activations re-using such arrays).

	Memory Footprint	n operations c
S_I	@$53 \cdot 86 \cdot 1 = 4558$	-
Pool0 (Weights)	-	-
S_I–Pool0 (Activations)	@$26 \cdot 43 \cdot 1 = 1118$	$2 \cdot 2 \cdot 53 \cdot 86 = 18232$
Conv1_00 (Weights)	364	-
Conv1_00 (Activations)	15652	391300
Conv1_01 (Weights)	4914	-
Conv1_01 (Activations)	15652	5.478.200
Pool1 (Weights)	-	-
Pool1 (activations)	@3822	4472
Conv2_00 (Weights)	4914	-
Conv2_00 (Activations)	@3822	1337700
Conv2_01 (Weights)	4914	-
Conv2_01 (Activations)	@3822	1337700
Pool2 (Weights)	-	-
Pool2 (activations)	@840	1092
FC Classifier (Weights)	2523	-
FC Classifier (activations)	@3	2520
Total	48933	8571216

5 Experiments results

The target device The considered tiny device is based on an ESP32 Microcontroller unit (MCU). Following the notation introduced in [9], we considered a RAM memory limit of $\bar{m} = 100$ KB, and set a limit on the execution time of the algorithm of 1 s. The device was used for both collecting the data and deploying the proposed solution.

Data description The input matrix S is characterized by M = 53, N = 200, each acquisition is W = 20 s long, and the frame rate was fixed to $f_r = 10$ Hz. f_l has been set to 1.66 Hz, such that the dimensions of S_I are M = 53 and L = 86.

Experimental results For the experimental results the dataset has been randomly split into 75% for the training and 25% for the testing, four runs have been considered (in a cross-testing fashion) and the average classification results is reported. Furthermore, the standard deviation was computed and used to estimate confidence interval (95% confidence).

Table 3 report the detailed memory footprint and the number of operations of the network on a per-layer basis, while table 4 describes the classification abilities of the proposed solution together with the memory footprint m and the computational load c for the subject-recognition scenario.

As a baseline we considered a simple algorithm that assigns the most represented class in the training dataset to every test data point.

Furthermore, since the imbalances in the dataset make the accuracy not the best metrics to evaluate the performance of the algorithm, the confusion matrix is here reported:

Table 4. Comparison of the results of the TyCNN-C and the baseline algorithm.

Network	Accuracy	m (kB)	c (10^6)
naive baseline	0.513	/	/
TyCNN	0.783 ± 0.076	47.79	8.57

True \Pred	absent (0)	child (1)	adult (2)
absence (0)	182 (82.72%)	34 (15.45%)	4 (1.83%)
child (1)	31 (19.01%)	113 (69.32%)	19 (11.67%)
adult (2)	0 (0%)	7 (15.22%)	39 (84.78%)

The proposed solution completely matches the technological constrains with $m = 47.8$ and $c = 8.57e6$. We measured experimentally the execution time of the solution on the ESP32 board. The total execution time is 940 ms, divided in 230 ms for preprocessing data and 710 ms to perform the inference with the TyCNN-C. Preprocessing required 27136 B to be executed in memory, and thus can be executed in the same memory space of dimension $\hat{m_a} = 31304$ B where the activations of the networks will be stored, hence not influencing the memory footprint.

6 Conclusions

The aim of this paper was to introduce, for the first time in the literature, a TinyML solution for subject-recognition based on UWB-radar. To achieve this goal we used TyCNN-C, an adapted version of the TyCNN network design used for presence detection. The effectiveness and efficiency of the proposed solution have been successfully evaluated on a real-world scenario for in-car subject recognition.

Future works will encompass comparisons with other state-of-the-art architectures, always-on scenarios for the proposed solutions, incremental learning mechanisms to support the on-device learning and the extension of the use of UWB-radar to human activity recognition.

Acknowledgment

The authors would like to thank Ing. P. Lento, and Dr. A. Bassi from Trusense s.r.l. and Ing. G. Viscardi from Politecnco di Milano for the support in the work.

References

1. J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.
2. C. Alippi, *Intelligence for embedded systems.* Springer, 2014.

3. C. Alippi, R. Fantacci, D. Marabissi, and M. Roveri, "A cloud to the ground: The new frontier of intelligent and autonomous networks of things," *IEEE Communications Mag.*, vol. 54, no. 12, pp. 14–20, 2016.
4. C. Alippi and M. Roveri, "The (not) far-away path to smart cyber-physical systems: An information-centric framework," *Computer*, vol. 50, no. 4, pp. 38–47, 2017.
5. A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," *arXiv preprint arXiv:2103.13630*, 2021.
6. J. Liu, S. Tripathi, U. Kurup, and M. Shah, "Pruning algorithms to accelerate convolutional neural networks for edge applications: A survey," *arXiv preprint arXiv:2005.04275*, 2020.
7. S. Disabato and M. Roveri, "Reducing the computation load of convolutional neural networks through gate classification," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.
8. T. Bolukbasi, J. Wang, O. Dekel, and V. Saligrama, "Adaptive neural networks for efficient inference," in *International Conference on Machine Learning*. PMLR, 2017, pp. 527–536.
9. M. Pavan, A. Caltabiano, and M. Roveri, "Tinyml for uwb-radar based presence detection," *Proceedings of WCCI 2022, IEEE*, Jul. 2022.
10. R. Sanchez-Iborra and A. F. Skarmeta, "Tinyml-enabled frugal smart objects: Challenges and opportunities," *IEEE Circuits and Systems Magazine*, vol. 20, no. 3, pp. 4–18, 2020.
11. R. David, J. Duke, and al., "Tensorflow lite micro: Embedded machine learning for tinyml systems," *Proceedings of Machine Learning and Systems*, vol. 3, pp. 800–811, 2021.
12. I. Fedorov, R. P. Adams, M. Mattina, and P. Whatmough, "Sparse: Sparse architecture search for cnns on resource-constrained microcontrollers," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
13. A. Kumar, S. Goyal, and M. Varma, "Resource-efficient machine learning in 2 kb ram for the internet of things," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1935–1944.
14. S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.
15. Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1389–1397.
16. Z. Cai, X. He, J. Sun, and N. Vasconcelos, "Deep learning with low precision by half-wave gaussian quantization," in *Proc. IEEE Conf. on computer vision and pattern recognition*, 2017, pp. 5918–5926.
17. S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *International conference on machine learning*. PMLR, 2015, pp. 1737–1746.
18. C. Alippi, S. Disabato, and M. Roveri, "Moving convolutional neural networks to embedded systems: the alexnet and vgg-16 case," in *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2018, pp. 212–223.
19. J.-E. Kim, J.-H. Choi, and K.-T. Kim, "Robust detection of presence of individuals in an indoor environment using ir-uwb radar," *IEEE Access*, vol. 8, pp. 108 133–108 147, 2020.

20. S. Chang and al., "An algorithm for uwb radar-based human detection," in *2009 IEEE Radar Conference*. IEEE, 2009, pp. 1–6.
21. A. R. Diewald and al, "RF-based child occupation detection in the vehicle interior," in *2016 17th International Radar Symposium (IRS)*, May 2016, pp. 1–4.
22. A. Caddemi and E. Cardillo, *Automotive Anti-Abandon Systems: a Millimeter-Wave Radar Sensor for the Detection of Child Presence*, Oct. 2019, pages: 97.
23. A. Rahman, V. M. Lubecke, O. Boric-Lubecke, J. H. Prins, and T. Sakamoto, "Doppler Radar Techniques for Accurate Respiration Characterization and Subject Identification," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 2, pp. 350–359, Jun. 2018, conference Name: IEEE Journal on Emerging and Selected Topics in Circuits and Systems.
24. T. Sakamoto, "Personal Identification Using Ultrawideband Radar Measurement of Walking and Sitting Motions and a Convolutional Neural Network," p. 9.
25. S. Rana, M. Dey, R. Brown, H. Siddiqui, and S. Dudley, "Remote Vital Sign Recognition through Machine Learning augmented UWB," in *12th European Conference on Antennas and Propagation (EuCAP 2018)*. London, UK: Institution of Engineering and Technology, 2018, pp. 619 (5 pp.)–619 (5 pp.). [Online]. Available: https://digital-library.theiet.org/content/conferences/10.1049/cp.2018.0978
26. K. E. Barrett and W. F. Ganong, *Ganong's review of medical physiology.* McGraw-Hill Medical, 2012.
27. J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.
28. B. Jacob, S. Kligys, and al., "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2704–2713.