POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

TinyML UWB-Radar based fall detection

Laurea Magistrale in Computer Science Engineering - Ingegneria Informatica

Author: Amedeo Carrioli

Advisor: Prof. Manuel Roveri

Co-advisor: Massimo Pavan

Academic year: 2022-2023

Abstract: Fall detection is the process of identifying human falls; this is an increasingly important task in various fields, in particular in healthcare and elderly care, where falls happen frequently. Falls can lead to important injuries, which often remain unnoticed for long periods of time, hence the need for an automated fall detection method. This project presents an algorithm that solves the problem of fall detection using an Ultra-Wideband (UWB) radar and an advanced neural network model, deploying it on an Internet of Things (IoT) device. The model, in fact, makes inferences directly on device, an Arduino microcontroller with limited resources. The model's training is computed on several UWB-radar recordings of falls and other human activities, from which it learns to recognize typical patterns of human falls.

UWB radar brings numerous benefits compared to traditional fall detection technologies, such as accelerometers, gyroscopes, and cameras. Indeed, it is a non-intrusive system that preserves the user's privacy (UWB radars do not capture or record visual images, individuals' faces, and physical appearances). An integral aspect of this thesis is the incorporation of Tiny Machine Learning (TinyML), a field of Machine Learning (ML) that focuses on the deployment of ML algorithms on low-power, resource-limited devices, such as microcontrollers. This ensures the compactness and energy efficiency of the proposed fall detection system.

To conclude, this thesis blends UWB radar, deep learning techniques, and TinyML to introduce a cutting-edge, privacy-centric solution for fall detection. This approach promises enhanced safety and sets a precedent for future developments in this delicate healthcare domain.

1. Introduction

Machine learning, in particular deep neural networks (DNN), has been successful in solving multiple challenges; DNN models proved to be more effective with large and high-quality training datasets. Tiny Machine Learning (TinyML) merges Machine Learning with compact and Internet of Things (IoT) devices, offering on-site data processing which ensures both privacy and minimized latency. Meanwhile, UWB radars are surging in popularity due to their versatile applications, including precise location tracking and motion detection with a single device. The fusion of TinyML and UWB radars opens the way to efficient, privacy-focused applications, exemplified by this work.

The aim of this thesis is to develop a DNN model able to make fall detection using UWB data and deploy it on an IoT device, making inference directly on-device. By doing so, the proposed solution is a low-power and low-memory consumption, on-device, privacy-preserving algorithm. To develop this algorithm, a novel neural network architecture, Fall-Net, has been designed, which is especially efficient for the UWB collected data. Indeed, none of the well-studied neural networks in literature were suited for this application given their architecture (too big and complex for deployment on IoT devices, a crucial component of this proposed solution). The results achieved are promising: 0.98 accuracy on the original model (no quantization applied) and 0.78 accuracy on the quantized model (which means the data types of the network are reduced from 32 to 8-bit) deployed on a microcontroller, and the memory footprint is only 44KB.

1.1. TinyML

TinyML is a field of machine learning focused on reducing the computational resources required for machine learning solutions. This allows such solutions to be deployed on limited-resources-embedded devices. These kinds of devices have been considered incompatible with ML solutions because of their limited memory and computational power; Thanks to model compression methods, however, machine learning algorithms can be successfully deployed on devices despite their limitation. For example, pruning of channels and layers of Convolutional Neural Networks (CNNs) has proven to be successful in reducing the memory and computational demand [4]. Another approach is quantization, which consists in using limited precision of data type, hence reducing the memory required to store CNNs models [1]. Importantly, these approaches apply to model evaluation only, which is the testing of an already trained model; The training of the model itself is a much more complex topic since it requires memory to store intermediate activations, and it relies on precise derivative calculations.

1.2. Fall detection

Fall detection is the process of identifying a person's fall. The time delay between a fall and the advent of medical assistance is crucial for the subject's health and must be minimized. Fall detection systems can be life-saving, especially in environments such as elderly homes where calling for help can be challenging.

1.3. UWB radar

Ultra-wideband (UWB) radars are a type of radar that exploit the benefits of low-power radio waves with expansive bandwidth (from 3.1 to 10.6 GHz) offering superior precision and imaging capabilities compared to traditional radar systems. Emitting brief pulses, UWB radars measure the time-of-flight taken for these pulses to reflect from objects, thereby calculating their position and hence detecting movements [3]. UWB radars are characterized by highly precise recordings (they can detect changes in the environment in the order of the mm), low energy consumption (typically < 0.1 W), and fast acquisition of data (each scan requires only some fraction of seconds to be collected). However, these systems can sometimes be sensitive to environmental obstructions. With ongoing innovations in the field, UWB radars are poised to establish new standards in real-time detection and safety applications [2].

2. Related works

Many fall detection systems have already been developed. These proposed solutions were initially categorized as wearable-device-based, ambient-sensors-based, and vision-based [10]. Wearable-based systems, employing accelerometers and gyroscopes embedded in devices such as wristbands and smartphones, are often intrusive for the individual; Ambient-sensors systems such as pressure sensors have problems with subject identification (who or what caused the pressure); Camera-based fall detection systems, lastly, have limitations concerning lack of privacy and high costs. Notably, Ozcan et al. [8] used wearable devices like smartphones and tablets, providing mobility in detection beyond controlled environments. Meanwhile, Kulurkar et al. designed a specialized low-power device with a three-axis accelerometer, achieving 95% accuracy [5]. UWB radar is a relatively new technology and it has been the subject of few studies, including one by M. Noori et al. [7], which used UWB radar data, collected through a robot and applied a long short-term memory (LSTM) neural network, achieving an impressive 99.6% accuracy. Another interesting study developed a fall detection system with UWB radar data in a single environment, utilizing a model with convolutional layers and convolutional long short-term memory [6], achieving a sensitivity of 95% and a specificity of 92.6% at a range of 8 meters.

Although all these solutions demonstrated that fall detection can be effectively automatized, none of them studied and developed a solution aimed at IoT, hence privacy-centric, low costs, and low computational requirement, and deployed it on a microcontroller with limited resources, such as the proposed algorithm.

3. Problem formulation

The primary objective of this thesis is to develop a neural network for fall detection using UWB radar data for IoT, and deploying it on a microcontroller with limited resources. More formally, this problem can be reformulated as the design of a classifier able to map a radar recording into its label: let $s_t \in \mathbb{R}^{N \times M}$, with $M, N \in \mathbb{N}$, be the signal received by the receiving antenna of the UWB radar, being N the number of scans or pulses emitted by the UWB radar and M the number of spatial "bins", which is the number of "quantized" distances in the acquisition range and $s_t \in S$, where S is the set of all radar acquisitions. Furthermore, $S[i, j]$ with $i \in \{1, ..., N\}$ and $j \in \{1, ..., M\}$ is the energy acquired by the receiving antenna at the i-th scan at the j-th bin. The problem aims to map s_t to its label y_t, where $t \in T$ is the set of labels, being $T = \{fall, non - fall\}$. In particular, the classifier has to occupy a memory M with $M <= A$ where A is the available memory on the microcontroller.

$$y_t = f(s_t) = \begin{cases} 0 & \textit{non-fall} \\ 1 & \textit{fall} \end{cases}$$

4. Device and constraints

Two devices have been employed for the development of the proposed solution: a UWB radar to collect the dataset and a microcontroller on which the solution has been deployed.

4.1. NXP Semiconductors UWB radar

The UWB radar used is produced by NXP Semiconductors [9], it works on UWB bands from 6.24 GHz to 8.24 GHz, and supports the detection and relative location of moving objects based on the changes in the reflected signal, measured by means of channel impulse response (CIR) estimates. A sequence of modulated pulses is transmitted and the receiver is continuously listening to any reflections from objects in the surroundings for the duration of the frame. The length of the computed CIR estimate is a function of the time taken for the pulse to reflect back from the object. The magnitude of the received signal is a measure of the strength of the signal reflected by the object and depends on the reflected object's properties such as size, material, and angle of incidence. Moving objects cause a change in the phase if the reflected carrier due to the Doppler effect.

4.2. Arduino nano 33 BLE sense

The microcontroller used, on which the proposed solution has been uploaded, is the Arduino nano 33 BLE sense, it has an ARM Cortex M4 MCU running at 64MHz and only 256KB of SRAM.

5. Proposed solution

The proposed solution is an algorithm that takes the UWB radar data as input, preprocesses them, and subsequently gives them as input to a pre-trained neural network model which classifies them as "fall" or "non-fall". More formally, let $s_t \in \mathbb{R}^{N \times M}$, $s_t \in S$, where S is the set of all radar acquisitions, and $M, N \in \mathbb{N}$, be the signal received by the receiving antenna of the UWB radar, being N the number of scans or pulses emitted by the UWB radar and M the number of spatial "bins", which is the number of "quantized" distances in the acquisition range. This signal is the input to a preprocessing function Θ_p, and its output, $\Theta_p(s_t)$ is the input to the classifier Φ which is composed by a feature extraction Φ_f block and a classification block Φ_c which classifies the input to the output class y_t, with $t \in T$ being T the set of classes, furthermore $T = \{fall, non - fall\}$. The constraints about Φ are imposed by the microcontroller for on-device implementation, in particular, the size of $\Phi + \Theta_p(S) <= M$, with M the memory available on the microcontroller.

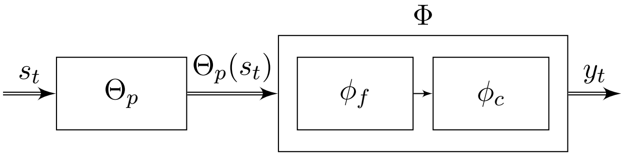

Figure 1: Proposed solution process

5.1. Preprocessing

The preprocessing of the radar data is essential to highlight the features of interest, such as the subject's movements. Each data sample S has initial dimensions of $256x128$. One could also view it as $128x128x2$ given that each pulse provides 256 values, comprising both amplitude and phase for every spatial bin. As part of the preprocessing, the norm of every value is determined, resulting in a 128x128 matrix, denoting time-space dimensions. Decluttering, a technique aimed to remove or reduce of unwanted interference is then applied.

In particular, "moving average filter" decluttering technique has been used for the proposed solution, although multiple other techniques have been tested. This technique works by computing the mean of a predefined range of data points and then offsets this average from the current data value, accentuating abrupt changes in value. This operates on both time and space dimensions, highlighting any substantial event that differs from its immediate average. Formally, it can be represented as:

$$M_{i,j}(R,w) = R_{i,j} - \frac{1}{w^2} \sum_{m=i-w}^{i-1} \sum_{n=j-w}^{j-1} R_{m,n}$$

where R is the radar matrix and $R_{i,j}$ represents the i^{th} time point at the j^{th} spatial bin, w is the window size, set as 3.

After decluttering, each recording is trimmed to capture the most pertinent details for the study, resulting in a more compacted $56x107$ matrix, where 56 denotes the space dimension (spatial bins) and 107 the time dimension (10.7 seconds).

5.2. Model architectures

Multiple architectures have been tested for this research, with two of them showing particularly good performances. Let's call the first one Fall-Net-2; Fall-Net-2 is a traditional style CNN with 15.297 parameters, it has an Input layer designed to accept 2D UWB radar data, represented as a $56x107$ 2D matrix. Then, the architecture has its feature extraction phase ϕ_f with a Conv1D layer that utilizes 8 filters of size 3 and the `'tanh'` activation function, followed by another convolutional layer which leverages 16 filters of size 6, this time with a `'relu'` activation function, enhancing the model's ability to extract more intricate patterns based on preceding layer outputs. Let's call the second architecture Fall-Net, which is the proposed solution; Fall-Net has 45.601 trainable parameters and its feature extraction part, ϕ_f, presents an "Inception Module", characterized by 4 parallel branches of different convolutional operations. Within this module, the 4 branches consist of a 1x1 convolution, a 3x3 convolution following a 1x1 convolution, a 5x5 convolution following a 1x1 convolution, and a 1x1 convolution following a max-pooling operation. These branches are then concatenated to form the module's output. The intention behind this parallel structure is to allow the model to learn different spatial hierarchies in the input data simultaneously. After the Inception module, the network flattens the output, passes it through a dense layer, includes a dropout for regularization, and finally outputs through a `sigmoid` activation function for binary classification, which is the classification block ϕ_c. Both models have been trained with Adam optimizer, learning rate scheduling with `ReduceLROnPlateau()` starting from 0.001 with a minimum of 0.0001, early stopping and `batch_size = 32`.

5.3. On-device deployment

Quantization, a process that uses limited precision data types, was conducted, reducing the model sizes to make them fit into the microcontroller. A full 8 bit quantization was introduced. The initial structure of the networks and size of the data was reduced drastically, from 32 bits to 8 bits, in particular, from `float32` to `UINT8`. Fall-Net-2 reduced its memory occupation from 68KB to 12KB, while Fall-Net from 186 KB to 44 KB. The models have been uploaded on the Arduino nano 33 BLE sense and inference has been run directly on the device. The time of execution for the prediction for each sample on device is 35 milliseconds for Fall-Net-2 and 19 milliseconds for Fall-Net.

6. Dataset

The dataset deployed, collected using the UWB radar model by NXP Semiconductors, which we will call NXP Semiconductors dataset, consists of radar recordings from various room scenarios, highlighting the algorithm's adaptability to different environments.

6.1. Data Collection Process

The dataset for this research was curated firsthand with assistance from several participants, with special attention on diversity in terms of movements, activities, and environments. Each recording lasted 12.8 seconds, producing 128 distinct radar pulses, since the working frequency of the radar was 10Hz. The radar was capable of capturing both the amplitude and phase of its 128 spatial bins since the signal returned as real and imaginary parts of a complex number. The data collected has a 128x128x2 matrix format, where 128 denotes the spatial bins and the number of radar pulses, and 2 represents the amplitude and the phase of each. After the first pre-processing step, which calculates the norms between the amplitude and phase of each complex number, the matrix has a shape of 128x128 (time-space dimensions).

The diversity of the dataset was further highlighted by the varied room environments it was captured in, each with its distinct architectural features and materials. Some rooms had unique challenges, such as the absence of a wall facing the radar, while others had potential interference sources, like glass and metal elements. The dataset consisted of 1,656 recordings, divided into eight distinct categories: `non-presence`, `sitting and moving around`, `standing and moving`, `chair still`, `standing still`, `fall`, `pick up something`, `laying on the ground`. To ensure the algorithm's versatility, the radar's position and orientation were frequently altered to prevent overfitting to specific room features. Activities were specifically chosen to represent the most common actions of individuals, particularly the elderly in nursing homes.

7. Experiments and Results

Multiple experiments have been conducted to find the best model performance while guaranteeing reduced network dimensions to fulfill the memory constraints of the microcontroller. As previously mentioned, multiple decluttering techniques have been tested, a comparison of these is shown in the next figure.

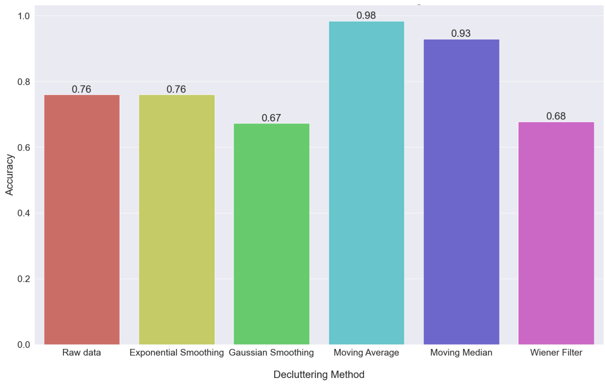

Figure 2: Fall-Net comparison over decluttering methods

Although the primary goal of this research is fall detection, experiments have initially been done for presence detection, a binary classification task on the same dataset where all the classes in which an individual is present have been given a `presence` label. Fall-Net reached 0.98 accuracy on the test set on the presence detection task and 0.74 accuracy using the quantized model on device.

The next table shows the results achieved for the fall detection task. The columns of the table represent, from left to right, the model name and decluttering technique, accuracy of the original model, accuracy on device using the quantized model, memory occupation of the model, execution time to invoke the model, and make inference on device. In particular, the row highlighted in green represents the proposed solution, which is Fall-Net employing moving average decluttering technique.

	Accuracy	Accuracy on device	Memory	Exec. time
Fall-Net + MA	0.98	0.78	44KB	19ms
Fall-Net-2 + MA	0.98	0.72	12KB	35ms
Fall-Net-2 + Raw data	0.94	0.76	12KB	35ms

Table 1: Fall detection results

Since the dataset presented 8 different activities, we extended the algorithm to make activity-type-detection (HAR), which is a multi-classification problem with 8 outputs. Fall-Net reached 0.65 accuracy. The quantization process made the performance drop, and this can be explained in multi-classification, `softmax` activation function is used, then, the `argmax()` among the probabilities is computed to find the output class. These probabilities are often relatively close to each other, and converting from `float32` to `UINT8` inevitably transforms similar floating point numbers into the same in numbers.

8. Conclusions

Operating within controlled environments, the developed algorithm shows the potential of TinyML combined with UWB radars as powerful tools for fall detection and beyond. The successful implementation and outcomes of this study underline the capabilities of this technology, not only in health monitoring and preventive care but extending its horizon across multiple domains.

References

[1] Zhaowei Cai, Xiaodong He, Jian Sun, and Nuno Vasconcelos. Deep Learning with Low Precision by Half-wave Gaussian Quantization. *arXiv e-prints*, page arXiv:1702.00953, February 2017.

[2] Dieter Coppens, Adnan Shahid, Sam Lemey, Ben Van Herbruggen, Chris Marshall, and Eli De Poorter. An overview of uwb standards and organizations (ieee 802.15.4, fira, apple): Interoperability aspects and future research directions. *IEEE Access*, 10:70219–70241, 2022.

[3] Davide Dardari, Chia-Chin Chong, and Moe Win. Threshold-based time-of-arrival estimators in uwb dense multipath channels. *IEEE Transactions on Communications*, 56(8):1366–1378, 2008.

[4] Song Han, Huizi Mao, and William J. Dally. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *arXiv e-prints*, page arXiv:1510.00149, October 2015.

[5] Pravin Kulurkar, Chandra kumar Dixit, V.C. Bharathi, A. Monikavishnuvarthini, Amol Dhakne, and P. Preethi. Ai based elderly fall prediction system using wearable sensors: A smart home-care technology with iot. *Measurement: Sensors*, 25:100614, 2023.

[6] Liang Ma, Meng Liu, Na Wang, Lu Wang, Yang Yang, and Hongjun Wang. Room-level fall detection based on ultra-wideband (uwb) monostatic radar and convolutional long short-term memory (lstm). *Sensors*, 20:1105, 02 2020.

[7] Farzan M. Noori, Md. Zia Uddin, and Jim Torresen. Ultra-wideband radar-based activity recognition using deep learning. *IEEE Access*, 9:138132–138143, 2021.

[8] Koray Ozcan and Senem Velipasalar. Wearable camera- and accelerometer-based fall detection on portable devices. *IEEE Embedded Systems Letters*, 8(1):6–9, 2016.

[9] NXP Semiconductors. Nxp semiconductors, 2023.

[10] Xinguo Yu. Approaches and principles of fall detection for elderly and patient. pages 42 – 47, 08 2008.