Exploration of Self-Learning Radar-based Applications for Activity Recognition and Health Monitoring

DOCTORAL THESIS

Gianfranco Mauro
Doctoral Programme in Information and Communication Technologies
Department of Electronic and Computer Technology
University of Granada
October 2023

Este documento está preparado para ser impreso a doble cara.

Exploration of Self-Learning Radar-based Applications for Activity Recognition and Health Monitoring

Directed by:
Prof. Diego Pedro Morales Santos
Prof. Manuel Pegalajar Cuellar

Developed at:
Infineon Technologies AG
BEX RDE RDF EET
Munich, Germany

Doctoral Programme in Information and Communication Technologies
Department of Electronic and Computer Technology
University of Granada

October 2023

Editor: Universidad de Granada. Tesis Doctorales
Autor: Gianfranco Mauro
ISBN: 978-84-1195-143-2
URI: https://hdl.handle.net/10481/89380

Acknowledgements

If you do not take risks, you can not create a future

Monkey D. Luffy

Choosing to go down the path of a doctorate is an ongoing exploration. It represents a journey in the continuous search for novel topics, self-organization, commitment, and motivation, as well as the state of the art. Over time, you realize that the most important goal is to believe in yourself. Daring and giving space and time to ideas. Learning from failures to rejoice in accomplishments. Personally, I would sum up the Ph.D. with the concept of "opportunity" rather than challenge. Opportunities that allowed me to learn more than ever before, always with the goal of overcoming the limits I faced.

I would first like to thank my thesis directors, Diego P. Morales and Manuel Pegalajar, for all the support they have offered me over the years. I would like to thank you most for the kindness and promptness with which you listened to my needs and offered help and advice. Thanks to you, I have learned that even in times of stress, kindness is always the most important thing to retain. Many thanks to Professor Vadim Issakov for all his precious advice and teachings, which have admirably helped me in the early stages of my Ph.D. research.

The Infineon Technologies AG IFAG BEX RDE RDF team deserves to be thanked. Here in this welcoming department, I have learned many things that I will cherish for the rest of my life. The mutual help among everyone allowed me to gather much experience in a short time. The fact that I have had both Ph.D. students and project managers around me has permitted me to learn a lot from the two different perspectives. I want to thank my supervisor, Kay Bierzynski, and my boss, Holger Schmidt. You both gave me, in addition to technical knowledge, many insights into the working world for my present and future. You were extremely helpful in understanding how to manage multiple tasks without losing sight of the overall goal. Many thanks also to the students who supported me during these years, Linyan and Youran. An important mention goes to the students whose theses I had the pleasure of

supervising, Ignacio and Maria. I learned so much from you all, even while trying to teach you what little I knew.

I would very much like to thank my colleagues here in Munich. Lorenzo, Ernesto, Hendrik, Miguel, Jessica, Arsalan, Mateusz, Julius, Jakob, Stefania, Moritz, Tommaso, Elfi, the Giovannis, and everybody else. From each of you, I learned something that I will keep with me. Thank you for all the moments spent together. To laugh and find joy even in the harsh moments of work and research.

I would like to thank my wonderful flatmates, Till and José. Thank you, José, for all the time spent talking about work or funny stories. With your experiences around the world, you have told and taught me a lot. And, Till, you have truly been a great reference point of mine over the years. Without even knowing it, you have given me so much will to achieve goals I did not even think to set. I would just like to say, "Keep it up". You are a wonderful person. Thank you to my lifelong friends as well as those who have been with me for a short time but will be with me forever. Thank you very much, Daniele, Mattia, Davide, Pablo, Carlo, Dario, Leonardo, Riccardo, and Salvo. I know very well that you are always there for me in moments of need.

Endless thanks to my family, both the relatively close family in Italy and the one further away in Canada. Many thanks to my uncles and cousins, who make me feel close from afar. Telling you about my adventures during this doctoral journey has definitely affected my motivation. Thank you so much to my brothers, Giuseppe and Elisa. You have always encouraged me to do my best. From you, I have truly learned so much. From your experiences, I understood how to face my challenges. From your qualities, I learned what to appreciate in life. Many thanks to my father, Fulvio. You passed on to me the love of technology and instilled engineering in me. I have always been fortunate to be able to ask you for both technical and human opinions. You instructed me to always keep calm and look at everything without thinking of anything to lose. Thank you very much to my mom, Angela. Inventiveness and tradition are your crafts. From you, I learned everything artistic that I have. Right during my Ph.D., I learned that design and art are fundamental parts of research. Imagination is the strongest of gifts.

In a very special way, I would like to thank my partner Sara, who is always with me in the hardest, most beautiful moments and at any moment. Thanks to you, I have strengthened the most important of my endowments, which unfortunately was fading: staying young at heart. Thanks to you now, I can still observe every little pleasure like a child, truly appreciating the gestures, caresses, or even a simple coffee. I owe you so much for that. Thank you for always making me feel alive and constantly discovering the world.

Finally, I would like to thank everyone else who is a daily part of my life and also anyone who will read this thesis. Thanks to all of you, I am the person I am, always pushing to give and research more.

Abstract

Sensor-based monitoring has proven effective in many settings for determining people's well-being and protecting their safety, even in difficult times like the COVID-19 pandemic. In many applications, radio wave-based systems are more versatile than those based on traditional sensors, thanks to non-contact sensing while preserving privacy. Health monitoring and assisted living are two good examples of how such systems are finding widespread usage in everyday applications. Good performance in such complex monitoring and recognition tasks is often achieved via machine learning. In particular, deep learning can aid with feature extraction, algorithm performance optimization, and forecasting. Yet, to learn how to tackle problems effectively, the generated models usually need access to a substantial amount of data. Furthermore, data preparation may be time-consuming and costly, especially when handled by specialists or when required in real-time systems. Few-shot learning techniques overcome these issues by adapting models to self-learn how to extract meaningful information from limited data. This is feasible by leveraging the learning context and previously acquired knowledge.

This doctoral thesis is the result of research on the exploration of few-shot learning techniques for radar-based applications in activity recognition and health monitoring. The investigation was performed by constraining the adaptation of radar-based solutions to limited data, ensuring the robustness of context generalization. The primary goal has been to investigate the use of limited data in very different non-contact applications, each with its own constraints and requirements. Millimeter-wave radar technology and few-shot learning have been used for hand gesture recognition, people counting, and human respiratory signal estimation. Such use cases, ranging from the millimetric displacements of vital signs to the distance of moving individuals, require specific information preprocessing. The generalization learning strategy has been explored for context and user adaptation while also accounting for preprocessing. Some of the algorithms were adapted to run on edge devices, allowing for end-to-end performance estimation and adaptation.

The research has been carried out under a doctoral contract at the facilities of Infineon Technologies AG, at its headquarters in Munich, Germany.

Resumen

La monitorización mediante sensores ha demostrado su eficacia en muchos entornos para determinar el estado de salud de las personas y proteger su seguridad, incluso en circunstancias difíciles como la pandemia de COVID-19. En muchas aplicaciones, gracias a la detección sin contacto preservando la privacidad, los sistemas basados en ondas de radio son más versátiles que los basados en sensores tradicionales. La vigilancia de la salud y la vida asistida son dos buenos ejemplos de cómo que son estos sistemas se están generalizando en las aplicaciones cotidianas. Un buen rendimiento en estas complejas tareas de supervisión y reconocimiento se consigue a menudo mediante el aprendizaje automático. En particular, el aprendizaje profundo puede ayudar en la extracción de características, la optimización del rendimiento de los algoritmos y la predicción. Sin embargo, para aprender a abordar los problemas con eficacia, los modelos generados suelen necesitar acceso a una cantidad considerable de datos. Además, la preparación de los datos puede llevar mucho tiempo y ser onerosa, especialmente cuando la manejan especialistas o cuando se requiere en sistemas en tiempo real. Las técnicas de few-shot learning resuelven estos problemas adaptando los modelos para que aprendan por sí mismos a extraer información significativa a partir de datos limitados. Esto es posible aprovechando el contexto de aprendizaje y los conocimientos adquiridos previamente.

Esta tesis doctoral es el resultado de una investigación sobre la exploración de técnicas de few-shot learning para aplicaciones basadas en radares en el reconocimiento de actividades y la monitorización de la salud. La investigación se ha realizado restringiendo la adaptación de soluciones basadas en radar a unos pocos datos, asegurando la robustez de la generalización del contexto. El objetivo principal ha sido investigar los usos de datos limitados en aplicaciones sin contacto muy diferentes, cada una con sus propios contextos y requisitos. La tecnología de radar de ondas milimétricas y el few-shot learning se han utilizado para el reconocimiento de gestos de la mano, el recuento de personas y la estimación de señales respiratorias humanas. Estas aplicaciones requieren un preprocesamiento específico de la información, que va desde los desplazamientos milimétricos de las señales vitales hasta la distancia debida a individuos en movimiento. Se ha explorado la estra-

tegia de aprendizaje por generalización para la adaptación al contexto y al usuario, teniendo en cuenta también el preprocesamiento. Algunos de los algoritmos se adaptaron para ejecutarse en dispositivos periféricos (edge), lo que permite la estimación y adaptación de las prestaciones end to end.

La investigación se ha realizado bajo un contrato doctoral en las instalaciones de Infineon Technologies AG, en su sede principal de Múnich, Alemania.

Contents

List of Figures

List of Tables

Part I

PhD Dissertation

Chapter 1

Introduction

You are the music while the music lasts

Thomas Stearns Eliot

Since the time of the earliest civilizations, human beings have always sought to use tools, energy sources, and materials for the necessities of life. As time progressed, innovations kept coming out faster and faster, which led to a sudden rise in the quality and expectancy of life. Instruments to measure wind speed or the time of day were already in use more than two thousand years ago. Physical parameter sensing enabled the prediction of events and potential hazards. In more recent times, such devices have taken on the name "sensing", enabling the detection of events or changes in environments.

Nowadays, sensors are used in the most varied applications, proving essential in the monitoring of industrial processes [1, 2], human activities, and vital functions [3, 4, 5]. Sensor-based systems can thus have very distinct goals with respect to application context.

Human Activity Recognition (HAR) is the process of classifying people's actions from measurements gathered by sensors [6, 7]. With increasing globalization and urbanization, HAR solutions can greatly benefit society in different ways. Data collected on the crowd can be used to estimate the number and monitor the actions of people, preventing disease contagion or hazards [8]. HAR-based systems are typically used to recognize common human actions. Such actions can set off alarms in critical contexts, such as elderly care [9]. Further, they can be used to activate and regulate domotic systems like smart homes [10]. Specific actions like hand gestures can also be leveraged in the context of Human Computer Interaction (HCI) [11, 12].

An important application field related to HAR is health monitoring. The estimation of an individual's vital functions can in fact facilitate the health status screening. Both short- and long-term monitoring can be effectively performed with various types of sensors [13, 14].

For all of the aforementioned applications, very robust solutions can be achieved by employing data from individual or combinations of sensors. Usually, sensor parameter combination or core information extraction is performed by computer vision or artificial intelligence (AI) techniques [15, 16, 17].

Depending on the context, however, some types of sensors may not be practically or ethically usable. Cameras, for example, could generate many privacy concerns for activity monitoring at home [18]. Wearable sensors can continuously collect information, but they can also lead to the recognition of activities not voluntarily tracked by the individual [19].

Among all the sensing technologies, radio-based sensors preserve more privacy thanks to their low resolution and set fewer user constraints thanks to their non-contact approach. Yet, the data obtained from such sensors are inherently difficult to interpret, requiring specific signal processing and prediction models. Especially for radio-based sensors, the nature of the features extracted limits the use of computer vision techniques. For this reason, AI and specifically Deep Learning (DL) are used in many applications to enable system monitoring and predict events [20, 21, 22]. Yet, many of the proposed solutions require training on a large amount of data to achieve robust performance. To counter this issue, a specific branch of Machine Learning (ML) called few-shot learning has become relevant in recent years [23]. Few-shot learning embraces several other ML sub-fields, including meta learning and active learning [24, 25]. Meta learning, often known as ”learning to learn“, refers to a set of algorithms whose primary objective is to learn how to approach new problems given prior experience, or meta-data [26, 27]. Active learning, on the other hand, aims to optimize model performance with as little labeled data as possible [28, 29]. The joint use of radar technology and meta learning branches can prove successful for HAR and Health Monitoring by ensuring privacy, no contact and rapid system adaptability, among other advantages.

1.1. Motivation

This thesis research is driven by a dual motivation. The first motivation is to investigate the manifold benefits of radar technology and how they can be leveraged to enhance activity recognition and vital sign sensing applications. The second reason is to explore the potential of few-shot learning algorithms in the chosen use cases for achieving robust performance and context generalization with limited data. By pursuing these two motivations, this research aims to enable the development of more accurate, reliable, and efficient radar-based systems, leading to new and innovative applications across diverse domains. The following two subsections analyze the two motivations individually.

Table 1.1: Comparison of Sensing Technologies features and constraints for Application-oriented scenarios.

Technology	**Cost per Unit**	**Power Consumption**	**Privacy Compliance**	**Scalability**	**Spatial Range**	**Non-Contact**	**Insensitivity to Environment**
Radar	✓	≈	✓	✓	≈	✓	≈
Camera	×	≈	×	×	✓	✓	×
CO_2 Sensor	✓	✓	✓	✓	×	✓	×
Infrared Sensor	✓	≈	≈	≈	×	✓	≈
LiDAR	×	×	×	≈	✓	✓	≈
Ultrasonic Sensor	✓	✓	✓	≈	×	✓	×
Wearable Sensor	×	✓	≈	✓	≈	×	≈
WiFi	×	×	≈	✓	≈	✓	≈

1.1.1. Harnessing the Benefits of Radar Technology for Activity Recognition and Health Monitoring

Many of contemporary solutions for activity recognition and vital sign sensing rely on technologies that require direct contact with the individual or lack of privacy [4, 30]. Other systems robustly tackle such issues but may have other disadvantages, such as limited data interpretability and weather dependence [31, 32, 33]. Sensors most commonly used for activity recognition and vital sign sensing include:

- Camera Sensor: cameras have been used for decades for a wide variety of non-contact monitoring and tracking applications. Red, green, and blue (RGB) cameras collect data in the form of multi-color pixel maps that can also be used to differentiate important visual features. Time-of-flight (ToF) cameras allow the processing of depth information by generating a 3-D representation of the collected data [34]. Further, the analysis of image sequences enables the estimation of multiple vital parameters. For example, depth information can be used to estimate an individual's breath rate. Color images can monitor eye movement and detect drowsiness, preventing fatalities in attention-demanding situations such as driving a vehicle [35]. Despite these advantages, the camera is ethically or practically incompatible with multiple use cases. Continuous monitoring of an individual can lead to serious privacy issues for recognition and tracking. This is especially critical if the data have to be saved on a server or used to improve an algorithm. Moreover, camera sensors are highly sensitive to atmospheric phenomena such as fog and low light. To counter this, in outdoor applications such as car parking, it is necessary to employ camera sensors with high spatial resolution, sensor fusion, and high system costs [36].

- CO_2 Sensor: this type of sensor is used to monitor processes that consume or produce carbon dioxide. The CO_2 is produced in the breathing process that characterizes living beings. As a result, carbon dioxide estimation can be used in non-contact monitoring solutions such as people counting in an environment [37, 38]. CO_2 sensors can also be placed in contact with an individual or applied transcutaneously. In that case, they can be used for measurements of metabolism or the health of the pulmonary system. The CO_2 sensor thus represents an excellent privacy-friendly and non-contact solution to tackle the estimation of some vital parameters and activities. On the other hand, however, estimation of many vital parameters is only possible through direct contact with individuals. Furthermore, counting people can only be done in closed indoor environments where CO_2 values are unaffected by other factors like ventilation.

- Infrared Sensor: this sensor perceives electromagnetic radiation with wavelengths longer than visible light. These frequencies, which are not visible to the human eye, convey the heat generated by objects and living things [39]. Thermographic cameras collect infrared radiation with an array of detectors, generating thermal images. Through thermal imaging, it is therefore possible, as opposed to using RGB cameras, to monitor people in low light or darkness. Low resolution thermal sensing further obviates major privacy concerns. [40]. Thermal imaging has numerous medical diagnostic applications, including monitoring body temperature and detecting joint inflammation [41, 42]. Despite all the advantages, infrared sensing is sensitive to all heat sources. Especially at low resolution, radiators, screens, or computers can make detecting people complex. In addition, thermographic cameras are generally dependent on thermal contrast, which makes the differentiation between objects and environments with similar temperatures difficult.

- LiDAR Sensor: this sensor makes it possible to determine the range of an object in relation to a series of transmitted laser pulses. The range is normally calculated as a function of the time it takes for the reflected light to return to the receiver. A 3-D scan is obtained by collecting range information in function of the direction and angle of transmitted pulses [43]. LiDARs do very well in monitoring applications, allowing tracking of targets even hundreds of meters away [44, 45]. In recent times, LiDARs have found applications for vital sign detection, such as in respiratory monitoring [46]. Despite its numerous advantages and being a non-contact technology, the LiDAR presents some remarkable disadvantages. Sensors capable of high resolution have a high cost per unit compared to camera or radar systems. High resolution in LiDARs is indispensable in applications such as vital sign sensing. Raindrops or other atmospheric phenomena can deflect transmitted laser pulses and make LiDARs unsuitable in several outdoor monitoring contexts.

- Radar Sensor: this sensor enables determining the range, radial velocity, and angle of arrival of targets in the field of view (FoV). The detection is achieved via radio waves emitted from a transmitting antenna. The information reflected from the targets in the FoV is combined with a transmitted signal reference. The resulting frequency-processed and analyzed information can be used to generate target information maps. Ranging, velocity (Doppler), and angle maps can be obtained and scaled to the resolution and maximum values that can be utilized [47]. A specific type of radar, called frequency modulated continuous wave (FMCW), further enables static target detection in the FoV by usually transmitting linearly frequency-modulated signals and receiving their reflections. This strategy enables the sensing of targets even in the ab-

sence of the Doppler effect (and therefore static) [48]. Recently, FMCW radars have found applications in a wide variety of industrial and even medical fields [49, 50]. This is thanks to the various advantages of the technology. Radars, in fact, enable non-contact sensing and are privacy-friendly because they do not allow to reconstruct distinguishing features of subjects. Furthermore, such technology is scalable, particularly for applications in the tens of gigahertz range (mm-wave), low cost and low power when compared to other radio frequency technologies like WiFi. Unlike LiDAR, radar systems are in general unaffected by weather phenomena, despite being frequently limited to applications in the tens of meters range. In fact, emitted radar signals, as a result of their high frequency, are rarely reflected by the widely spaced raindrops. Yet, raw radar data are very often difficult to interpret with classical computer vision tools. As a result, many radar solutions use deep learning to extract relevant features [51]. Moreover, because of their many advantages and versatility, radars are used extensively in activity recognition, especially in short-range applications [52]. Some examples of the research focus are hand gesture recognition [53] and people tracking [54]. Thanks to the micro Doppler effect [55], some radar technologies can also be used to analyze periodic displacements in the millimeter range. In this way, the breath and heart signals of an individual positioned in front of the board can be estimated [56].

- Ultrasonic Sensor: this sensor estimates the distance of a target by measuring the time needed for a transmitted signal to get reflected and reach the receiver (ToF). Compared with radar, the ultrasonic transducer senses ultrasonic sound waves through piezoelectric crystals or capacitive micro-machined ultrasonic transducers. Thanks to the non-contact/non-invasive uses of this technology, ultrasound has already been used for a long time in a wide variety of applications. This technology is particularly expendable for very short-range applications up to 1 m. In fact, it is used for biometric [57] or gesture recognition in HAR [58, 59]. Furthermore, ultrasound is widely used in medicine for diagnostic purposes [60]. In sonography the distinct absorption coefficients of different tissues are leveraged to create a structural image of internal organs. High-resolution ultrasonic sensors and multi-receiver systems are also employed in the field of vital sign sensing [61, 62]. Despite its many advantages and applications, ultrasound also has considerable disadvantages. Ultrasound waves require a propagation medium like air or tissues. The speed of sound waves in air is highly dependent on temperature and humidity, which can lead to significant measurement errors as the data collection context changes. Additionally, the high attenuation of ultrasound in air prevents its convenience at ranges of more than a few meters.

- Wearable Sensors: systems that can be worn are becoming widely popular for a wide range of uses. Wearable as belts, bracelets, or watches, many solutions allow for continuous monitoring of an individual's movements and physiological parameters [63, 64]. Thanks to their high performance and non-invasiveness, they are often used as standard reference sensors for the development of various non-contact systems. Many activity monitoring solutions use information fusion between accelerometers, gyroscopes, and magnetometers. A fusion of the data collected by these sensors allows improved and instantaneous estimation of acceleration, rotation, and orientation [65, 66]. For vital sign sensing tasks, the sensors vary greatly with respect to the specific target parameter [67]. The electrocardiogram (ECG) sensor measures the electrical activity of the heart by placing electrodes on the skin. The photoplethysmography (PPG) sensor measures changes in blood volume and can be used to monitor heart rate, blood pressure, and oxygen saturation. Breathing belts measure the respiratory signal in newtons, and wearable temperature sensors can be used to monitor fever or changes in body temperature due to exercise or stress. Wearable sensors have several advantages, including the ability to track multiple parameters simultaneously and enable continuous monitoring in a wide variety of contexts remotely. On the other hand, continuous monitoring of sensitive information can lead to privacy problems. In addition, wearing the sensors can be uncomfortable in the long run for the user.

- WiFi sensing: using the WiFi signals that a router is constantly transmitting, it is possible to detect the presence, activity, and even vital signs of people who are present within the coverage area. WiFi sensing is based on analyzing the signals that are partially reflected off of people when they move or breathe. The changes in signal amplitude, frequency, and phase in the reflected signals are often processed using machine learning algorithms to extract features that correspond to specific activities or vital signs [68, 69]. WiFi technology has the great advantage of enabling these measurements in a completely non-contact mode. In addition, it does not need image capture and is widely available in indoor environments. This brings the great advantages of privacy preservation and ease of availability and scalability. On the other hand, WiFi consumes much more power (W) than other radio frequency techniques, such as radar (mW). This is because WiFi sensing requires continuous signal transmission and reception, which is not required for radar. For radars, in fact, signals are typically transmitted in short bursts, and the receiver only needs to listen for return signals, resulting in lower power consumption.

Specific tables that outline the strengths and weaknesses of sensing features and target applications can be generated in relation to all of the considered sensing technologies.

Table 1.1 lists the characteristics of the analyzed sensors for various technologies, features, and application constraints. Table 1.2 presents specific applications where one of the sensors under consideration can be fully or partially employed. In both tables, the checkmark (✓) represents an advantage for a particular technology in a particular category, a cross (×) represents a disadvantage, and a double tilde (≈) represents a case-dependent feature or only partially addressable applications.

In Table 1.1, it can be seen that radar does not have a major disadvantage in any of the categories under consideration. The price is much lower than that of a camera, LiDAR, or many wearable systems. The same is applicable to compliance with privacy, scalability, and non-contact use. Radar requires more power than ultrasonic and CO_2 sensors on average. Such sensors, however, are highly constrained by context. In fact, radar turns out to be usable both indoors and outdoors and has little dependence on atmospheric or environmental phenomena. From the application point of view in Table 1.2, radar sensing, with the proper configurations and application constraints, turns out to be versatile in a wide range of use cases. Wearable sensors are highly expendable in many use cases, but they are constrained by wear requirements and, in some cases, cost. WiFi sensing or ultrasonic sensors have similar versatility to radar but show numerous disadvantages when compared with it (see Table 1.1).

Based on technological features and application constraints, radar sensing proves to be an excellent and versatile choice for many use cases. Thus, radars can be used to widely explore adaptability needs in activity recognition and vital sign detection cases.

1.1.2. Leverage limited radar information via Few-shot Learning

Despite its versatility in a variety of applications, radar sensing has inherent complexity due to its data structure. In fact, raw radar data often needs several steps of application-dependent preprocessing to extract useful features. In addition, the presence of ambient noise and the infeasibility of fully reconstructing the shape of targets limit the use of classical computer vision techniques.

For the listed reasons, DL techniques are normally used to extract useful features from radar data [70, 52]. In cutting-edge solutions, DL is used on partially preprocessed data and, in some cases, even on raw data [71]. Ordinarily, though, a large amount of training data is necessary to achieve adequate performance in a given application. Further, when the classical DL

Table 1.2: Comparison of Sensing Technologies for Various Applications.

Technology	Activity Recognition	Entrance Detection	People Counting	People Tracking	Gesture Sensing	Breath Sensing	Heartbeat Sensing	Other Vital Signs
Radar	✓	✓	✓	≈	✓	≈	≈	≈
Camera	✓	×	✓	✓	✓	≈	×	≈
CO_2 Sensor	×	≈	✓	×	×	✓	×	×
Infrared	≈	✓	✓	≈	✓	≈	≈	≈
LiDAR	✓	≈	✓	✓	✓	×	×	≈
Ultrasonic	≈	✓	≈	≈	✓	≈	×	≈
Wearable	✓	×	✓	≈	✓	✓	✓	✓
WiFi	≈	✓	✓	≈	✓	≈	≈	≈

learning framework is adopted, it is hard to robustly adapt a model in new contexts without the collection of a large amount of new data. This approach is especially unsuitable in use cases where rapid and continuous adaptation of contexts or users is required.

Table 1.3 presents the main setup parameters and achieved performance of some state-of-the-art radar-based solutions. All methods taken as examples need a large number of users and training examples to achieve optimal performance in a given context. In [72], a total of 5,019 hours of recordings have been utilized for training and validating the model to achieve a robust 99 % accuracy rate on hand gesture recognition. Despite the various DL topologies used, which very often leverage temporal information, context adaptation is critical and leads to a significant performance drop [73, 74].

Meta and active learning techniques, leveraging prior information, can enable context for user adaptation with no or only a small drop in performance. With few-shot learning and the meta learning episodic approach, a model can learn to generalize for a set of tasks instead of specializing in just one. This method also yields, in many applications, robust generalizations to unseen contexts [26, 27]. Active learning can be used to improve model performance if new labeled data is available. One approach is to use only examples with the highest classification uncertainty, and thus potentially the most new information content [28, 29].

In the radar application frame, few-shot learning can enable models to self-learn how to extract features from newly performed hand gestures. A meta learning approach can be used to train a breath rate estimation model on a new user or to count the number of people in a new environment. If the radar is deployed to a new location in an environment, active learning can be used over time on newly available data to boost performance.

Few-shot learning is thus a useful tool to counteract the need for large amounts of data and complexity in context generalization of radar applications.

Despite the advantages of few-shot learning, a system exposed to continuous and rapid changes may require real-time adaptations to new contexts. Consequently, it is also important to assess what the potential limitations of adapting the few-shot learning approach might be in the various use cases. For a given system, such analysis is done in terms of average adaptation time and prediction time on new samples. Moreover, many state-of-the-art solutions with radar-enabled machine learning can run even on processors with limited performance, i.e., they can be Edge deployed [75, 53]. This topic is still little explored in the context of meta learning and active learning algorithms for radar applications. One of the main challenges is, in fact, adapting the model to the Edge without losing generality, thus ensuring robust performance in new contexts.

1.2. Objectives

The objectives of this doctoral thesis are related to exploring the use of few-shot learning for radar applications, aiming at context and user generalization. Contributions to the state-of-the-art include the study of generalization methods for radar-based use cases, the preprocessing of radar data, the algorithmic optimization of active and meta learning methods, and the study of deployment and inference estimation of developed solutions. These contributions can be summarized as follows:

1. **Context generalization for radar-based applications research**
    - Research about potential context generalization needs in radar-based use cases.
    - Analyze specific datasets and elaborate tasks generation strategies for the episodic learning approach.
2. **Radar data processing research**
    - Research on radar data preprocessing strategies for efficient feature extraction.
    - Design of optimized deep learning topologies for radar data and few-shot learning.
3. **Meta and active learning algorithmic optimization**
    - Research and optimize current meta learning algorithms for radar-based applications.
    - Define new meta learning algorithms and strategies aiming at radar-based context generalization.
    - Investigate a common evaluation framework to assess different meta learning strategies and experiment performance.
    - Explore and define active learning strategies for task fine-tuning.
4. **Inference and edge implementation research**
    - Analyze inference and adaptation trade-off of the generalization models.
    - Explore potential edge implementations of radar-based applications, leveraging few-shot learning

Table 1.3: Examples of state-of-the-art for deep-learning based FMCW radar solution features. Datasets size, approach and achieved accuracy.

Application	**Authors**	**Model**	**Number of Classes**	**Participants**	**Dataset Size**	**Achieved Test Accuracy**
Gesture Recognition	Zhang et al. [76]	Recurrent 3-D CNN	8	4	4,000	96 %
	Choi et al. [77]	LSTM	10	10	4,000	98.48 %
	Hayashi et al. [72]	CNN	8	7,647	558,000	99 %
Activity Recognition	Saeed et al. [73]	ResNet	6	99	4,510	85–96 % [a]
	Helen et al. [78]	Tower CNN	6	56	4,052	87.58 %
People Counting	Stephan et al. [74]	DCNN and KD	7	≥ 6	67,600	89–58 % [b]
	Servadei et al. [79]	DCNN, KD and LAR	6	≥ 5	95,000	83.0 %
Breath Rate Sensing	Gong et al. [80]	Adaptive LSTM	Regression	14	≈ 500 [c]	3.32 bpm (avg. error)

LSTM stands for long short-term memory. ResNet is the residual neural network. DCNN is deep CNN.
KD is knowledge distillation. LAR is label-aware ranked loss, and bpm is beats per minute.
[a]The accuracy drops from 95 % to 85 % when actions are performed by multiple subjects in various geographical spaces.
[b]The accuracy drops from 89 % to 58 % when test data are recorded with another sensor with a different orientation and aspect angle.
[c]Each instance represents a data session lasting 30 seconds. The number of instances collected is not clearly specified.

1.3. Outline

The presented document constitutes a thesis by way of a compendium of publications. Therefore, the most significant findings from the doctoral program research serve as the foundation of the thesis.

The thesis consists of two main parts:

- **Ph.D. Dissertation**: this part describes the objectives, methodology and achievements of the carried out research. The section 2 presents the adopted methodology according to the objectives outlined in the section 1.2. The section 3 summarizes the main achievements with respect to the state-of-the-art. The section 4 presents the conclusions of the conducted research and lists some potential future trends.
- **Publications**: this part gathers the journal articles pertinent to the thesis dissertation research. These publications are devoted to the aforementioned research aims.

The publications part consists of four publications. Respectively, three indexed articles in scientific journals and one conference paper:

- Gianfranco Mauro, Mateusz Chmurski, Muhammad Arsalan, Mariusz Zubert, and Vadim Issakov. "One-shot meta learning for radar-based gesture sequences recognition." In Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part II 30, pp. 500-511. Springer International Publishing, 2021.
- Gianfranco Mauro, Mateusz Chmurski, Lorenzo Servadei, Manuel Pegalajar Cuellar, and Diego P. Morales-Santos. "Few-shot user-definable radar-based hand gesture recognition at the edge." IEEE Access 10: 29741-29759, 2022.
- Gianfranco Mauro, Maria De Carlos Diez, Julius Ott, Lorenzo Servadei, Manuel P. Cuellar, and Diego P. Morales-Santos. "Few-Shot User-Adaptable Radar-Based Breath Signal Sensing." Sensors 23, no. 2: 804, 2023.
- Gianfranco Mauro, Ignacio Martinez-Rodriguez, Julius Ott, Lorenzo Servadei, Robert Wille, Manuel P. Cuellar, and Diego P. Morales-Santos. "Context-Adaptable Radar-Based People Counting via Few-Shot Learning." Applied Intelligence, 1-29, 2023.

Chapter 2

Methodology

All truths are easy to understand once they are discovered; the point is to discover them.

Galileo Galilei

This section presents the methodology used to achieve the objectives presented in Section 1.3. Specific methodologies for each of the defined sets of objectives are presented in the next sections.

2.1. Research on radar-based applications and context generalization

The advantages of radar technology are described as one of the two main motivations behind this doctoral research in Section 1.1.1. Radars are versatile in several use cases thanks to their properties. Among the many advantages, this technology enables non-contact tracking of targets while ensuring privacy compliance, scalability, and very low influence from environmental variables. The first step of research in this area for the doctoral program has been to focus on the choice of a specific radar technology and modulation. This has been necessary to define the set of radar-based use cases to be considered for the overall research. In fact, distinct types of radars may be particularly effective in automotive or industrial applications but not employable in other use cases, such as vital parameter estimation. The choice of using millimeter wave, frequency modulated continuous wave (mm-wave FMCW), was made after consulting numerous surveys on the topic. These surveys cover both the technology, such as [81, 82] and the potential applications, such as [47, 20, 51]. The employed radar board has been the 60 GHz mm-wave Infineon XENSIV™ DEMO BGT60TR13C.

The use cases to be taken under examination have been chosen for a variety of reasons. Specifically, the range of applications, the characteristics of the information to be processed, and the complexity of adapting cutting-edge solutions to new contexts. Regarding the application range, it has been chosen to explore use cases with different distance magnitudes, specifically mm, cm, and m, so as to fully exploit the properties of the mm-wave FMCW radar. The signal amplitude, frequency, and phase have been the main information considered from the gathered radar information. This allowed for broader research on the need and type of context in which a developed solution may have to generalize. For both of these reasons, the choice of applications has been made by analyzing surveys on the potential use cases of the mm-wave FMCW radar technology [47, 51]. For different use cases, the context or scenario in which the generalization of the solution may be needed can considerably vary. Some adaptation needs may, for example, involve a new user with related characteristics, a new environment, new actions, or new functional parameters of the device. The study of the limitations of state-of-the-art solutions for different use cases has been used to understand where context generalization is often crucial and not easily achieved. In this case, specific surveys related to application branches of radar technology have been analyzed [53, 83, 84, 85] as well as specific solutions that clearly point out the limitations of context generalization [75, 74, 86].

For all the reasons mentioned, the doctoral program research had the main focus on the following radar-based use cases (Figure 2.1):

- Hand gesture recognition (cm)
- Breath signal sensing (mm–cm)
- In-door people counting (m)

The data collection approach for the various use cases, taking into account contextual variation, is presented as part of the Achievements in Section 3.1. The results obtained on the various context generalizations are collected in all the publications attached to the thesis [87, 88, 89, 90].

2.2. Research on radar data processing and optimized deep learning topologies

In this research, emphasis has also been given to radar data processing with respect to the various applications, which are discussed in Section 2.1. The first objective in this regard has been to investigate what data processing techniques are used for radar across use cases. This has been carried out by consulting surveys related to radar data processing for the various use cases, such as [20, 85, 83]. The usual preprocessing approach is to apply

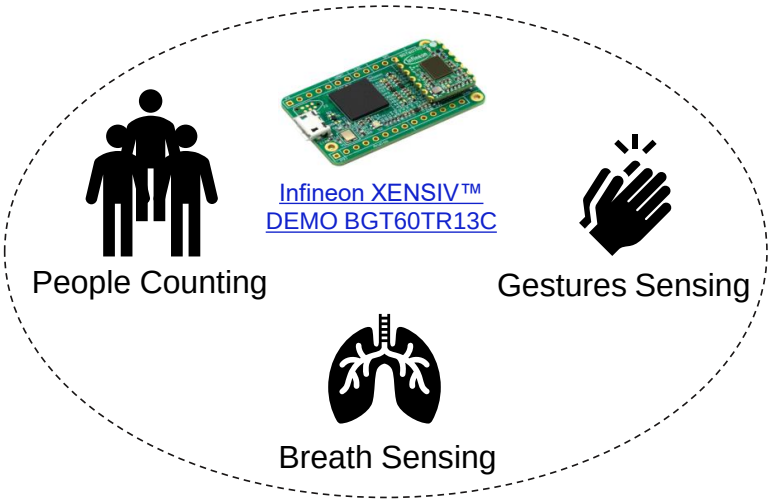


Figure 2.1: Use cases that were the focus of the research. On top in the image is the Infineon XENSIV™ DEMO BGT60TR13C, used as the mm-wave 60 GHz radar board for all applications.

filtering and transformation techniques to single samples or sequences of data to extract useful information and patterns. The main characterizations evaluated for single samples have been range-Doppler maps and range-angle maps. For sequences of data, representations that yield a large reduction in data dimensionality without important loss of information have mainly been evaluated. Specifically, the range, velocity, and angle over time, as well as the unwrapped phase signal over time spans.

In many cutting-edge approaches, the extraction of features often involves the use of ML. As mentioned in the introduction, Section 1, radar data are inherently complex to interpret. Compared to ordinary images obtained through a camera, for example, radar data cannot be directly processed with computer vision techniques to locate targets. This is mainly due to the resolution of the sensor and the nature of the sensor itself, which is normally developed for target tracking rather than segmentation applications. With feature extraction architectures such as Convolutional Neural Networks (CNNs), ML can be leveraged for both data handling and task solving. Other architectures, such as autoencoders, are efficiently pretrained on a set of available data and then used as the backbone of models to reduce the dimensionality of features. Few works have also associated radio-frequency data processing with the episodic learning approach typical of meta learning and few-shot learning [91, 92, 93]. This is of particular importance for this doctoral research since the use of few-shot learning to leverage limited context information represents one of its two main motivations (Section 1.1.2). In general, representations with low dimensionality or no time dependence are chosen for task resolution. This is a common approach to simplifying the

computation needs and the task of feature extraction for the chosen models. For all papers that are part of this cumulative doctoral thesis [87, 88, 89, 90], it was chosen to use non-recurrent data handling techniques. Hence, it has been chosen to simplify the preprocessing of the temporal component of the data by mapping the time information into single channel representations.

The metrics used for assessing the preprocessing techniques are the number of model parameters and the task performance achieved with varying feature sizes.

2.3. Research on meta learning algorithms

As outlined in the introduction Section 1, meta learning represents a relatively new but broad branch of ML aimed at task generalization via some prior collected experience. Given the variety of types of meta learning algorithms, a consultation of the main available surveys on meta learning [26, 27] has been conducted. The choice of the meta learning algorithms to be further investigated has been conditioned by the chosen radar-based use cases as defined in Section 2.1. For each of the specific applications, the state-of-the-art has been researched, focusing on major work oriented toward context generalization, such as [92, 74]. The main branch of meta learning chosen for research has been *optimization-based* meta learning [94, 95]. In this class of algorithms, information from one context or task is shared with another by using gradient descent or averaging the weights of neural networks. The main advantage of these approaches is their *model-agnosticism*. This property makes them suitable for generalization purposes, regardless of the architecture of the neural networks. This valuable attribute uncouples the data handling and processing needs dictated by radar-based applications from the chosen training algorithm. Thanks to the *model-agnosticism*, the research focus has been broadened to the optimization of neural network topology for feature extraction and Edge deployment (Sections 2.2 and 2.5). Optimization-based algorithms have been investigated, adapted, and optimized in three of the four papers that are part of the compendium of this thesis. Respectively, as classification for solving the hand gesture recognition tasks in *One-shot meta learning for radar-based gesture sequences recognition* [87] and *Few-shot user-definable radar-based hand gesture recognition at the edge* [88] and as regression for the breath signal sensing task in *Few-Shot User-Adaptable Radar-Based Breath Signal Sensing* [89].

Another class of meta learning algorithms chosen for investigation are relation-based [96, 92]. These approaches enable powerful context generalization by learning to classify combinations of instances rather than individual samples. This feature, though, is inherent to the model topology, which extracts features from test images by comparing them with representative samples from the training classes. The relation-based algorithms

have been mainly investigated, adapted, and optimized for the people counting task in the paper *Context-Adaptable Radar-Based People Counting via Few-Shot Learning* [90]. This paper also investigates the potential fusion of optimization-based and relation-based techniques by trying to leverage the advantages of both. The results in this regard are described in Section 3.5 and detailed in the paper. The differences between *optimization-based* and *relation-based* meta learning, are shown graphically in Figure 2.2.

The approaches presented in [90] have been further evaluated on Omniglot [97]. Omniglot is a public dataset typically used for assessing the performance of few-shot learning experiments. This dataset contains a total of 1,623 handwritten characters from 50 different alphabets around the world.

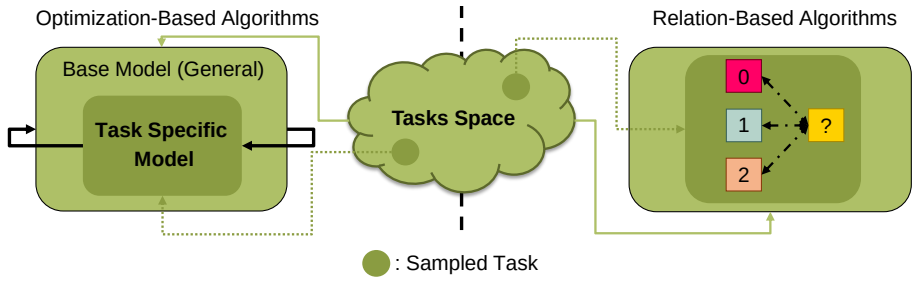


Figure 2.2: The *optimization-based* algorithms (left), propagate the information obtained on a task to the base model via weighted parameter summation or gradient method. The *relation-based* algorithms (right), try to learn the relationship between data rather than its class. In this case, a particular ML model architecture is required.

Many of the cutting-edge meta learning solutions employ their own evaluation method for contexts and class generalization [94, 95, 96]. Unfortunately, this does not allow for an objective evaluation of the advantages of one approach over others, especially in various model classes. Regardless of the algorithm, this doctoral research also had the main goal of developing a common framework for evaluating models developed with various algorithms. The main goal has been to ease the comparison of various algorithms by generating numerical and graphical evaluations in the same format. The results of this research are outlined in Section 3.6 and detailed in all publications that are part of the compendium [87, 88, 89, 90]. The evaluation framework permits the comparison of various algorithms not only in terms of episodic generalization performance but also in terms of the trade-off between single-sample inference time and training time to adapt to a new task (Section 3.8). Most of the meta learning experiments have been performed on TensorFlow with Python as the programming language. The general metrics for evaluating the solutions were the performance (accuracy and loss) of the models with different algorithms, time to adapt to a new task, and single-sample inference time.

2.4. Exploration of active learning strategies for episodic learning

Active learning is also a relevant branch of ML that, as introduced in Section 1, aims to optimize the model's performance with as few labeled instances as possible. Like meta learning, active learning can be leveraged to generalize to a new context by filtering out useful information from a pool of new available data [98]. For the same reason, active learning can be further used for fine-tuning the performance on a task of a ML model [29]. In this doctoral research, the use of active learning has been tied to the use of meta learning in radar-based use cases. Specifically, try to understand whether prior experience learned through episodic meta learning can be beneficial for active learning fine-tuning. Thus, given prior information on other similar tasks, the goal has been to understand whether, for adaptation to a new context, active learning can better filter out relevant data than starting from a random initialization of parameters. The idea is that, if a new task is generated from the same domain used for meta learning, the uncertainty estimation can leverage the whole domain knowledge acquired by a chosen model during the episodic training.

For this research objective, the achieved results are detailed in the publication *Context-Adaptable Radar-Based People Counting via Few-Shot Learning* [90], which is part of the compendium of papers. Major achievements are also discussed in Section 3.7.

The main evaluation metric for active learning has been the performance of the model under varying formulations of prediction uncertainty and types of initialization of neural network parameters (random or after meta learning).

2.5. Exploration of meta learning implementations at the Edge

Meta learning, as mentioned in Section 2.3, represents a powerful tool for context generalization when new data are available. This approach might prove useful to counter typical problems in deploying ML solutions in industrial environments, such as data drift [99]. Nevertheless, the deployment and adaptation of meta learning solutions at the Edge can be complex and sometimes prohibitively expensive due to computation requirements in custom neural network topologies. In other cases, however, the agnosticism of meta learning training approaches, such as the optimization-based methods outlined in Section 2.3, can be very advantageous in enabling deployment at the Edge. Research on the main neural Edge devices available on the market has been done to choose which one to employ. As this is an exploration, to

test the feasibility of the deployment of meta learning as a proof-of-concept, it has been chosen to test the deployment on the Intel® Neural Compute Stick 2. This neural accelerator, compared to many other devices, allows the deployment of many ML operations without strong limitations on maximum computation and Floating Point Operations (FLOPS). The main results obtained are presented in the paper *Few-shot user-definable radar-based hand gesture recognition at the edge* [88], which is part of the compendium of publications in this thesis. The main achievements are also detailed in Section 3.9.

The main metric for evaluating deployment at the edge was single-sample inference evaluation and adaptation training on new tasks as the available computation units (Intel® Core™ i5 Processor, Raspberry® Pi3 ARM microprocessor, and Intel® Neural Compute Stick 2) changed.

Chapter 3

Achievements

Do not judge me by my successes, judge me by how many times I fell down and got back up again.

Nelson Mandela

In this chapter, the main achievements of the doctoral program, which led to the publications attached to this dissertation, are discussed. The results obtained, accomplished in multiple areas, are presented in specific chapter sections according to the main objectives defined in Section 1.3 and elaborated through the methodology described in Chapter 2. Each of the following sections in this chapter presents the achievements of one or, normally, several of the attached publications. Reference is also given in each section to the specific publications that are part of the compendium and contain relevant information. The projects acknowledgment section details the European funding projects that have funded the research performed and the results achieved during this doctoral program for Infineon Technologies AG 3.10. A final collaborations section 3.11 cites all the related co-authored publications during the PhD program. These cited publications have been the result of collaborative projects with research and industrial partners.

3.1. Research about context generalization in radar-based use cases

This section outlines the main achievements of context generalization for the selected radar-based applications, as defined in Section 2.1. For each of the papers that constitute the compendium of publications, a specific meta-dataset in relation to the defined context generalization requirements has been generated. Each created meta-dataset attempted to collect the most heterogeneous information possible in as few data samples as possible so

as to correctly evaluate the potential of the meta learning generalization approach (Section 2.3). According to the radar use case under investigation, the need for generalization in new scenarios may vary considerably. The following paragraphs provide specific use case descriptions of the context variations taken into account in the compendium's publications. A graphical representation of the scenario variations investigated in this research for the various use cases is shown in Figure 3.1.

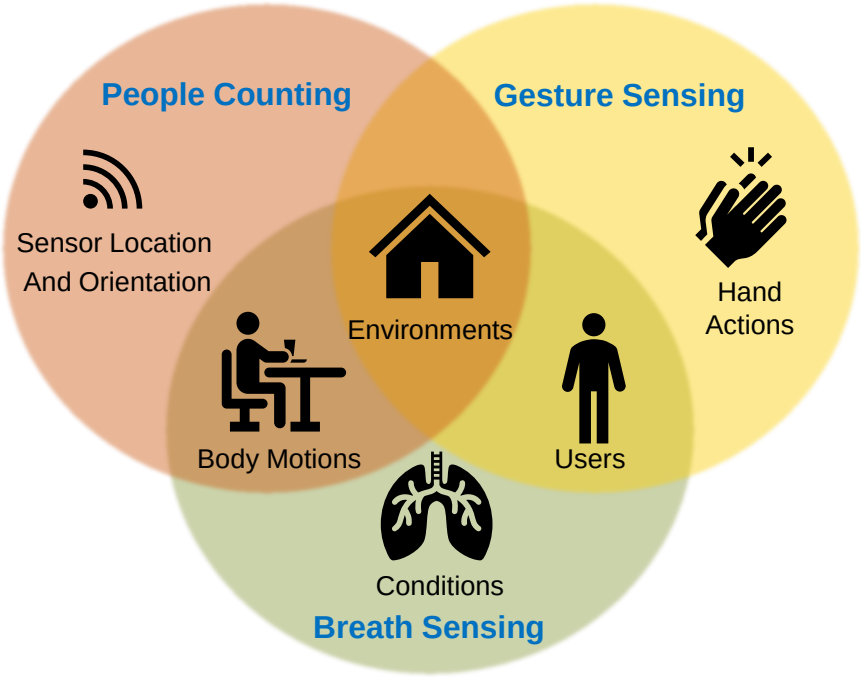


Figure 3.1: Investigated scenario changes with respect to use cases. Environmental variations, such as different rooms or offices, have been considered in all applications. Different individuals have been distinguished for the gestures and breath sensing tasks. The motion of parts or the whole body of individuals mainly influenced the tasks of people counting and breath sensing. Positioning and orientation of the radar sensor have been considered for the people counting task. A person's vital condition exclusively influenced breath sensing data collection. In contrast, actions performed with hands influenced the data collection for gestures.

Both *One-shot meta learning for radar-based gesture sequences recognition* [87] and *Few-shot user-definable radar-based hand gesture recognition at the edge* [88] focus on the task of radar-based hand gesture recognition. For such a use case, the main context variations involve the type and mode of gestures performed, the users, and the environment in which they are gathered. The radar information requires proper preprocessing (Section 3.3) to ease the characterization of similar types of gestures, such as the actions

of *push* and *slide* in a given direction. Extracting the whole range, velocity, and angle-of-arrival information for each gesture may be inadequate for some HCI systems. The need for all these features can render a gesture too complex to be performed for some categories of people, such as users with motor impairments. For this specific reason, the meta-dataset generated for *One-shot meta learning for radar-based gesture sequences recognition* [87], contains only four types of gestures with very distinct features. However, four actions can be far from a sufficient number of commands in most HCIs. Consequently, the idea has been to consider not single actions but sequences of gestures as single classes to increase interfacing possibilities. For the paper *Few-shot user-definable radar-based hand gesture recognition at the edge* [88], a more inclusive meta-dataset has instead been generated in order to further test the limitations of the episodic meta learning approach. In this case, 20 classes of gestures with 20 examples per class performed by 5 people in 2 environments have been gathered. For this larger number of gestures, the information on range, speed, and angle of arrival must be correctly extracted to simplify differentiation. The various users were also asked to perform the gestures according to their definition, given just a general indication of the specific gesture. Every gesture has been performed up to 40 cm from the radar board and within a time span of roughly three seconds.

The publication *Few-Shot User-Adaptable Radar-Based Breath Signal Sensing* [89] focuses on user generalization for a radar-based solution of respiratory signal estimation at a workplace. The main goal of this research has been the development of a solution that is user-adaptable and can promptly generalize to new vital patterns. In this case, data have been collected from 24 users at two distances and in two offices. The Infineon XENSIV™ 60 GHz mm-wave radar has been used for the recordings, while a breathing belt has been used as a reference. Ten sessions of 30 seconds each per user have been collected. Different kinds of factors have been taken into account for the generation of the meta-dataset for such a task. The main context adaptability requirement for such an application is dictated by the user's characteristics. Users can, in fact, have very different breathing patterns in relation to their age, physique, health, and other attributes. As non-contact, usersáctions during recordings can cause corruption of radar data, so-called motion corruption. Many cutting-edge solutions assume that users remain idle in front of the radar [86, 100]. In this case, no restriction on stillness has been given to users, and they have been allowed to talk, laugh, or use the keyboard during data collection. Further information on how that information is handled is outlined in Section 3.3 and detailed in the publication [89] and. The board orientation as well as the distance from the board to the tracked subject have also been taken into account for the recordings. Such characteristics have, in fact, an impact on the intensity of reflections and therefore the estimation of vital signs for the recorded data.

The task of in-door people counting has instead been handled in the paper *Context-Adaptable Radar-Based People Counting via Few-Shot Learning* [90]. People counting by radar enables a scalable, low-cost, and privacy-friendly solution. Yet, most cutting-edge solutions are highly dependent on the context in which they are deployed [74]. Deep learning-based solutions can achieve high performance in a given environment, but they can perform poorly in other contexts due to multiple factors. One of the main factors is the structure of the training environment and the presence of static objects. Data variation must involve multiple environments with different characteristics, such as room size, furniture, windows, or other objects that cause reflections. The variation must also involve the orientation of the radar sensor, which has to be moved to multiple locations for the gathering of new data sessions. In addition, data must also be collected on various people, moving at various speeds and in various directions, sitting or standing still under certain circumstances. For this use case, data have been recorded in three different indoor environments with different furniture and sizes. Each session involved 0 to 5 people in a room at the time, placing the radar in different positions and orientations. Ten different people took part in the recordings, with varying recording times between 60 and 90 seconds. The recorded data have been used to generate three distinct meta-datasets to enable better evaluation in different contexts. One dataset called *Mixed-Dataset* contained both training and test data from sessions collected from all environments but from different orientations of the radar board. The other two meta-datasets used two rooms for the training dataset and the third room as the test (the smallest and largest environments, respectively).

For all the use cases, the information on training and test split and task generation is related to the task generation strategy, and it is therefore detailed in Section 3.2.

3.2. Research on tasks generation strategies for the episodic learning approach

Once the meta-datasets have been defined for the various use cases (Section 3.1, it has also been necessary to define a task generation strategy for the episodic meta learning approach (Section 2.3). In principle, what should be the context in which a model should be able to generalize for each new training episode? The number of classes (ways) and training examples (shots) for each experiment are also important parameters for task generation. The data availability and number of classes per episode can drastically vary the complexity of the experiments. Further, it is also important to correctly divide the generated meta-datasets into training classes and test classes. In fact, the models must have sufficient heterogeneous information available to ensure generalization in training but also sufficient test information in order

to adequately evaluate the behavior in new contexts.

In *One-shot meta learning for radar-based gesture sequences recognition* [87], the permutations of the four available gestures were used in sequences to generate tasks, creating sixteen classes. All classes containing the 'Left/-Right' gesture were exclusively considered part of the test data. Experiments were performed in 2–, 4– and 5– way, with 1-shot only. As the number of ways varied, the examples representing the tasks were randomly sampled from the set of available example sequences.

In *Few-shot user-definable radar-based hand gesture recognition at the edge* [88], the 20 available gestures in the meta-dataset have been divided into 12 gestures of training and 8 of testing. This division has been done randomly, ensuring only that opposing actions such as left swipe and right swipe were part of the two separate sets to ensure greater generalization and better evaluation. For this meta-dataset, the whole information of range, velocity, and azimuth angle of arrival over time has been elaborated for every single gesture. This intrinsically allowed for high variation among gestures. Therefore, a much broader set of tasks has been sampled compared to the four gestures scenario. Experiments have been conducted in 2–way and 5–way, with 1–shot, 3–shot and 5–shot. To achieve high generalization, the training tasks have been randomly sampled as a combination of the available training gestures. The evaluation has been performed in the same manner over test gestures.

In *Few-Shot User-Adaptable Radar-Based Breath Signal Sensing* [89], every task focuses on the single user adaptation. Every episodic task adaptation utilizes randomly sampled recorded sessions at different distances between the sensor and the tracked user. Data containing more motion corruption are proportionally less important than others for signal estimation. This is handled via preprocessing and deep learning, as outlined in Section 3.3. Being a single-user regression and reconstruction problem, the experiments have all been performed at 1–way for 1–, 5– and 10–shot.

In *Context-Adaptable Radar-Based People Counting via Few-Shot Learning* [90] the episodic tasks have been differently generated for the training and evaluation steps. The training has been performed by randomly sampling classes belonging to various environments, sensor orientations, and numbers of people in a room for a given session. The evaluation, instead, has been done so that a specific number of people occupy specific label positions in a ranked manner. For counting up to 3 people, for example, the first output class of the model has been 0, the second 1, the third 2, and the last 3. Features had been sampled from the various available environments, but with a raking constraint on the defined ways. The ranking has been a necessity for the intrinsic goal of generalizing by counting people in an environment rather than distinguishing various environments or orientations.

3.3. Research on radar data preprocessing strategies for efficient feature extraction

Radar information is handled in very different ways as the use case and application range changes. For mm-wave radars, the range of action is often limited to a few meters [101]. For applications in the meter range, such as people counting, the intention is usually to extract and track targets rather than their features. In the cm range, thanks to a higher resolution in speed and range, it is also possible to track the features of the movements. For the recognition of hand gestures, oscillations due to actions like tickling or rubbing can be tracked. The estimation of vital parameters requires the analysis of displacements in the range of mm or a few cm. In this case, the information is all encoded in the phase signal of the radar data, which is very sensitive to such displacements, especially if periodic. Regardless of the application, the information extracted from radar data may be too large in size to be directly classified by fully connected neural networks. Specific DL operations suitable for feature extraction have been employed in this research as part of preprocessing to make the information more easily handled in the learning and prediction phases. This also made it possible to reduce the number of parameters to train for the models, simplifying episodic, few-shot adaptation without harming generalization to new tasks.

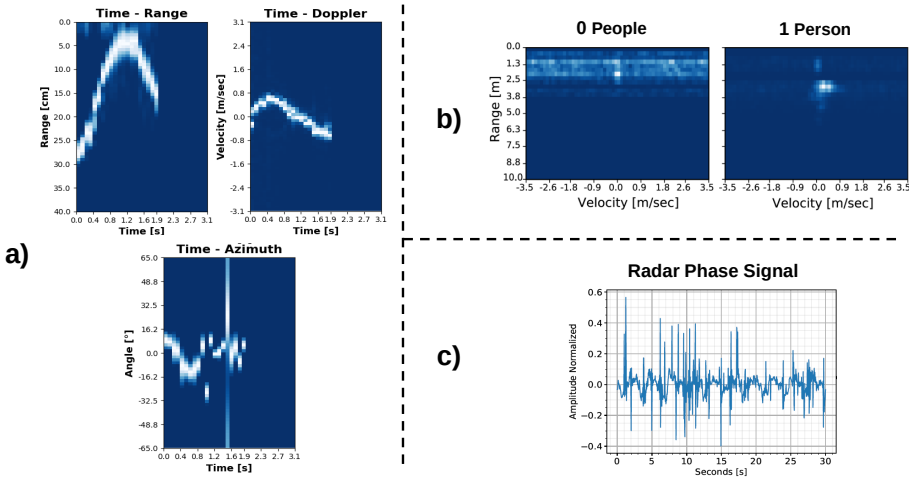


Figure 3.2: Temporal representations of a hand gesture, used in [87, 88], are shown in (a). Shown in (b) are two examples of range Doppler maps used as system input for the indoor people counting task [90]. The unfiltered radar phase signal collected in a breath sensing session for [89] is shown in (c).

In *One-shot meta learning for radar-based gesture sequences recognition* [87], as pointed out in Section 3.1, only data from four very distinct gestures have been used. To simplify the potential use of the system as an HCI and

also reduce power consumption, only range information collected from a single radar receiver channel has been preprocessed. Each radar frame collected in the gesture time window was frequency processed to extract the range information. To simplify the time dependence of the collected data over time, only the peak range information was used and projected along an axis. This made it possible to generate Range Time Maps (RTMs). By preserving temporal information in a very small dimensionality, RTMs are widely used in the literature [53, 75]. For each sampled task, RTMs represent instances to be fed to a neural network as channels (gesture sequences). Further information processing is done episodically through a CNN, which extracts key information through episodically trained convolutional filters.

In *Few-shot user-definable radar-based hand gesture recognition at the edge* [88], given the higher complexity of the collected gesture set, the information of velocity and azimuth angle over time in addition to range has also been used. Velocity and angle information have been extracted by frequency analysis and Capon beam-forming [102], respectively. This information is then also projected in time by generating the Doppler and angle time maps (DTM and ATM). The use of all this information as channels for every gesture sample grants the ability to distinguish feature-wise different kinds of motions with particular oscillations and patterns.

In *Few-Shot User-Adaptable Radar-Based Breath Signal Sensing* [89], a tailored preprocessing pipeline extracts, for every given session, the phase information containing the user's breathing profile. The raw data would be too heavy and full of irrelevant information to be directly fed into a neural network for breath profile extraction. This would also limit the capability of the few-shot episodic approach given the very limited information available per user. The pipeline smartly keeps track of the distance of the person from the radar and extracts phase information only for the few range bins corresponding to the position of the user. This feature reduces at the same time the computation cost of the pipeline and the potential noise caused by environmental reflections. The main innovation of the presented pipeline is the estimation of the motion corruption level due to user movements. This is done purely in terms of signal processing, autocorrelating the extracted breath signal with itself over each data session for dissimilarity computation. Whenever the dissimilarities are greater than a given threshold, the motion corruption flag is triggered. The amount of corruption is used to optimize the breath signal estimation in the episodic deep learning phase. Higher is the corruption, more the term for only central frequency information is relevant (it would not make sense to map the corrupted radar signal to the belt data). This consents to overcome the major limitation of idle users of many cutting-edge radar-based solutions [86, 100]. For this use case, a Convolutional Variational Autoencoder (Conv-VAE) is completely embedded into the meta learning training for episodic feature extraction (Section 3.4).

The application of a double digital bandpass filter to the radar data is only done after the meta learning step to extract the breath profile.

In *Context-Adaptable Radar-Based People Counting via Few-Shot Learning* [90], the preprocessing aims at increasing the signal-to-noise ratio on sequences of available radar frames. For every new raw time frame, frequency processing is performed so as to extract the range and velocity profile of moving people in the environment. The signal is mediated over the available receiving channels. Specific window filtering and mean subtraction are also performed to reduce the level of noise in the samples. A moving average is performed over a buffer of recorded data to further enhance the SNR. Such averaged frames, in the form of range-Doppler images, are used as input into the Weighting-Injection Net (Section 3.4) for the two different training strategies chosen (Section 3.5). A graphical depiction of the preprocessed radar data for various applications is provided in Figure 3.2.

3.4. Design of optimized deep learning topologies for radar data and few-shot learning

Standard radar-based solutions with ML can be hard to generalize to new contexts due to the complexity of feature patterns and scenario specifications (Section 2.1). Specific DL architectures can be designed to gather the most information possible about contexts and learn to compare examples in the same scenario rather than just using previously learned information. Good topologies for generalization can, anyway, be bulky and require a huge number of parameters to achieve good performance. Some targeted, cutting-edge modules aiming at feature extraction and comparison can help find a good trade-off between task generalization and model size.

In *Few-shot user-definable radar-based hand gesture recognition at the edge* [88], a Conv-VAE enabled the reduction of data dimensionality while retaining the information's content. A graphical illustration with respective formulation of the cost function for a Variational Autoencoder (VAE), is shown in Figures 3.3. Once the meta-dataset has been generated (Section 3.1), the Conv-VAE has been pretrained over training classes and used as a backbone. The Conv-VAE performance has been compared in the paper with a CNN. The CNN reached slightly higher values of intra-task accuracy but with an order of magnitude more trainable parameters. The model, which used the Conv-VAE as its backbone, also required half of the adaptation time for new tasks. With 5–way 5–shot, the accuracy loss is just around 1 %, from 94.19 % to 92.97 %. The episodic optimization further fine-tuned the encoder layers of the Conv-VAE for feature extraction. Further, this topology has been fully deployed at the Edge on the Intel® Neural Compute Stick 2 (Section 3.9). In the few-shot learning frame, the encoder part of the Conv-VAE is concatenated into a sequence of dense layers for task training. Such

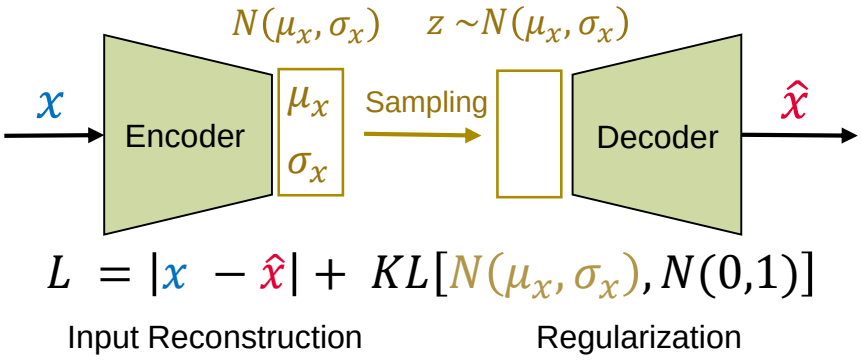


Figure 3.3: In general, a VAE consists of an encoder a decoder and a custom cost function. The encoder compresses the input x into the latent space z. The decoder approximately reconstructs the input, generating $\hat{x}$, from the z representation. The representation generated in z consists of a set of mean values μ_x and standard deviations σ_x corresponding to normal distributions $\mathcal{N}$. The loss function (L) tries, on the one hand, to minimize the differences between $\hat{x}$ and x. On the other hand, a regularization term makes sure that the various μ_x and σ_x are constrained to a normal distribution $\mathcal{N}$. This is possible by minimizing the Kullback-Leibler (KL) divergence between the normal $\mathcal{N}(\mu_x, \sigma_x)$ and the unit distribution $\mathcal{N}(0, 1)$ with mean value 0 and standard deviation 1. Both encoders and decoders can have convolutional layers for feature extraction (Conv-VAE).

a procedure makes it fully employable in the few-shot approach and also deployable at the edge thanks to the pretty straightforward topology.

In *Few-Shot User-Adaptable Radar-Based Breath Signal Sensing* [89], a specific two-branch topology has been developed for user-wise episodic adaptation (Section 3.2). The employed algorithm for episodic training has been MAML [94]. The designed topology consists of a Conv-VAE, which maps, in the training phase, the preprocessed phase information extracted from radar data to the reference belt signal. The other branch constrains the features generated in the Conv-VAE latent space to the central frequency of breathing for the given session. The ground truth of such information is extracted from the recorded belt data. The overall topology enables the episodic extraction of the most relevant breathing information with around 740,000 trainable parameters. A custom loss function has been designed for the task's accomplishment so that:

1. The model can promptly adapt to new users given the limited number of sessions available.

2. The motion corruption information is leveraged session-wise for model learning, avoiding exploiting corrupted information in a harmful way.

If the corruption is low, then priority is given to reconstructing the breath signal using the belt data as a reference. If the corruption is high, then the extraction of the central frequency of breathing is the main priority. Such an approach is proportional to the level of corruption. The prediction on the test sample is also a contribution from the two terms. The performance of the designed C-VAE topology as a function of feature size has been a crucial aspect to examine. This has been analyzed as a function of the latent space dimension, which corresponds to the dimensions of the retrieved features. The latent space values used for the MAML 1–shot experiments ranged from 16 to 128. A latent dimension of 32, which was also used for all of the episodic experiments, led to the lowest mean loss value. It appears that a latent dimension of 16 is insufficient to fully extract from the radar phase all important breathing information. At the price of twice as many parameters, a dimension of 64 resulted in similar average loss values to those with a feature size of 32. Both a latent space of 64 and a latent space of 128 lead to a discernible decrease in precision when looking at the values at 95 %. This indicates that the higher-dimensional extracted features have a tendency to overfit the training set and do not transfer well to test subjects.

In *Context-Adaptable Radar-Based People Counting via Few-Shot Learning* [90], the employed topology is a variation of the Weighting Network presented in [92] and it is part of the domain of relational net models [96] (Section 2.3). The proposed topology, called Weighting-Injection Net, employs an injection module in place of the classic embedding module for dimensionality reduction. The injection module increases the input data dimensionality while generating a feature-enriched representation of support and query samples for the subsequent relational phase. Such a topology requires more parameters than embedding-based topologies but leads to higher accuracy when the number of shots per task increases. The Weighting-Injection Net topology has been tested with the traditional meta learning training approach of the Weighting Network [92], on both the recorded radar data for in-door people counting (Section 3.1) and the public dataset Omniglot [97]. Compared to the embedding module, the injection operation projects the features in a larger dimension and allows the extraction of more information for the following comparison module. In addition, this specific characteristic reduces the possibility of overfitting on specific tasks during episodic learning. This allows for greater generalization in multiple of the defined experimental setups. Mainly, the increased dimensionality of support and query data via the injection module facilitates the extraction of important features with higher filter dimensionality when many shots per class are available (Section 3.2). The main results obtained with the Weighting-Injection Net are outlined in Section 3.5.

3.5. Research on meta learning algorithms for radar-based use cases

As specified in Section 2.3, it was chosen to further research *optimization-based* and *relation-based* algorithms in the meta learning branch.

Optimization-based algorithms present *model-agnosticism* among their main advantages, but can also be complex to train for generalization due to the large number of hyperparameters and the chosen task generation strategy (Section 3.2). Dependence on the chosen neural network architecture and computational complexity can also lead to training instability, resulting in performance collapse [103].

In *Few-shot user-definable radar-based hand gesture recognition at the edge* [88], different strategies aiming at increasing the stability of the Model Agnostic Meta Learning (MAML) [94] algorithm are presented. The designed approaches, suitable for optimization-based meta learning, were tested on the radar task of hand gesture recognition and also used together with some techniques introduced in the MAML++ algorithm in [94]. The baseline algorithm, consisting of MAML and stabilization techniques borrowed from [94] has been called $^{+}$MAML and used for comparisons. Respectively, the presented methods are Dynamic Meta Class Weighting (DMCW), Task-Specific Gradient Clipping (TSGC), and Evaluation-Based Gaussian Noise Summation (EGNS). DMCW aims at reducing specific class overfitting in episodic learning. The labels of some evaluation examples are predicted after every task adaptation. Such labels are used for the computation of class weights for each task adaptation epoch. The obtained weights are also used for the intra-episode adaptation. TSGC limits the magnitude of the gradient updates in every task update when it exceeds a certain threshold. Instead, no gradient clipping is used in the intra-episodic adaptation. This approach assures that the generalization update will be generally more relevant than the individual tasks. EGNS has the purpose of avoiding meta-overfitting, which means depending too much on tasks sampled from the classes of training while losing power of generalization in the test classes. When the validation accuracy on tasks sampled from training exceeds a certain threshold, some Gaussian noise is added to the next training examples to increase the complexity of the training and avoid meta-overfitting.

The three presented methods are more or less efficient than the baseline, according to the few-shot experiment setup and the employed topology and number of parameters (Section 3.4). Specifically, with a large number of parameters (CNN), the DMCW does not lead to performance improvements since, with more trainable parameters, the model understands better differences between classes without simply overfitting the few samples available for just one or a few classes. DMCW is even more counterproductive as the number of shots increases. The designed strategies are particularly effective

when only a very few examples per class are utilized since they counter the problems of task- and meta- overfitting. The DMCW introduces benefits for the model using the encoder of the Conv-VAE. The proposed methods lead to better results than all the state-of-the-art methods except Weighting Net [92] for most of the performed experiments. Yet for the model with Conv-VAE, the number of required parameters is half that required by Weighting Networks, making it easier to deploy on edge devices and resulting in lower inference costs by varying the number of shots.

Compared with optimization-based algorithms, relation-based networks [96, 92] require significantly fewer hyperparameters but may suffer from adaptation lack and instability with only a few shots per class available [104].

In *Context-Adaptable Radar-Based People Counting via Few-Shot Learning* [90], a new algorithm called Model-agnostic meta-weighting (MAMW) combines the two-step strategy MAML [94] for task and intra-episodic adaptation with the robustness and fast adaptation of relation-based meta learning. The MAMW algorithm targets mainly to increase the stability in training of 1– and 2–shot experiments employing the Weighting Network [92] as the network topology. In such an architecture, the representative examples per task of the training classes are called support examples. Instead, the examples to be mapped to a given label are called query examples. The Weighting Network's intrinsic capability of instance comparison makes it a strong episodic learning method. However, this strategy exhibits learning instability with only a few training shots per episode. This is because, particularly in 1-shot learning, the query is compared to each unique support instance, which may not be sufficiently class descriptive. As a result, MAMW utilizes the task adaptation step of MAML to compare the available support examples via the Weighting Network with a noisy version of themselves. This leads the model to extract the most possible information from support examples before the outer step is evaluated on the query. The MAMW algorithm has been tested on both the recorded radar data for in-door people counting (Section 3.1) and the public dataset Omniglot [97]. In the context of people counting via radar, the comparison of the range Doppler of support with a noisy version of themselves has the further advantage of reducing the training dependence of a given context (Section 3.1). In practice, the task adaptation focuses on learning the main information about the targets contained in the range Doppler images rather than potential noise due to the static environment, a.k.a. clutter. The 1- or 2-shot experiments carried out with the MAMW yield higher average accuracy values than the Weighting-Injection Net with a pure relation-based training approach, regardless of the meta-dataset employed (Section 3.2 and Section 3.4). In these particular situations, the initial comparison with a noisy version of the support samples, which emphasizes the possible inherent noise of the query data, allows MAMW to

provide the model with more information. As the number of shots increases given the fixed number of ways, the classical relation-based approach applied to the Weighting-Injection Net guarantees better performance except for the *Mixed-Dataset*. As the number of available support instances increases and adaptation occurs in totally new contexts, MAMW may cause a shift in the learning goal toward noise detection rather than learning the query class. This is due to the step of comparing the multiple available support examples with a noisy copy of themselves. Both the relation-based training of the Weighting-Injection Net and MAMW led to better performance than state-of-the-art in all the performed people counting experiments. The same exact results for both MAMW and the Weighting-Injection Net can be observed for all the experiments performed on the public dataset Omniglot.

3.6. Investigation of an evaluation framework to assess the performance of the meta learning experiments

As outlined in Section 2.3 and Section 3.5, meta learning algorithms can approach the task of context generalization with very different strategies. Even in the same class of algorithms, such as optimization-based algorithms, the use of different task adaptation or intra-episodic strategies can make the comparison of achieved performance more challenging. The evaluation becomes even more complex when other classes of algorithms are considered, such as relation-based, where query data are directly compared to the available support instances.

A common evaluation framework aiming at assessing and correctly comparing the performance of types and classes of meta learning algorithms, as specified in Section 2.3 has been elaborated in this research. The framework aims at assessing generalization in a validation phase after each episodic adaptation on randomly sampled training and test tasks. A further evaluation is performed at the end of the episodic adaptation on a set of randomly sampled test tasks to evaluate the performance after the final adaptation.

In *One-shot meta learning for radar-based gesture sequences recognition* [87], a common evaluation framework for assessing the classification performance of optimization-based algorithms is presented. The evaluation is performed after each episode on both a task sampled from the meta-dataset of training and the meta-dataset of tests. The sampled instances are utilized to tune the model to the specific task. The learned weights are only utilized for the specific task and forgotten right after, without influencing the episodic adaptation. Some evaluation samples are instead used in the following prediction phase to estimate the accuracy. For a fixed evaluation strategy, as the episodes progress, the overall evaluation accuracy for new

tasks should increase for both training and test classes, thanks to the acquired experience from past tasks. The performance is assessed over sequences of episodes, generating box plots over the obtained accuracy values. With episodes progression, box plots should converge to 100 % accuracy, while the specific whiskers and quantiles should get narrower. A final evaluation is done on meta-test sampled tasks after the last generalization episode. In this case, the average value of accuracy is used as a Key Performance Indicator (KPI). A schematized version of the accuracy evaluation framework is shown in Figure 3.4. Such a strategy allows for a accurate comparison of the various state-of-the-art algorithms.

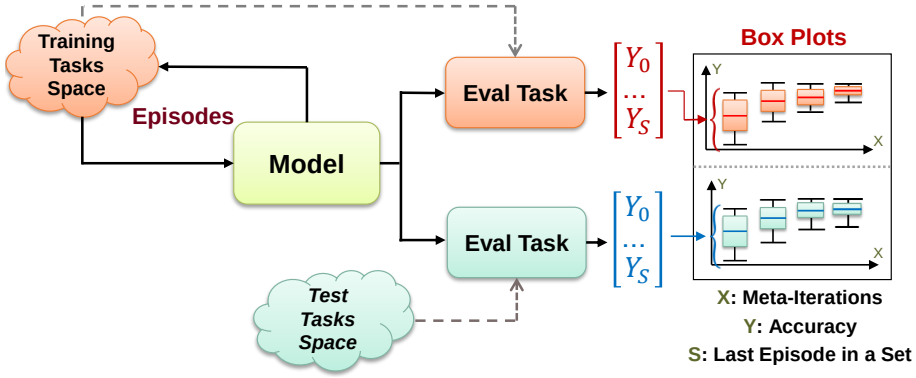


Figure 3.4: Evaluation framework for assessing generalization accuracy over a set of tasks. After each episode, an evaluation is performed on tasks sampled from the training and test meta-datasets respectively. The performance is assessed over a set of episodes, building accuracy box plots. The specific whiskers and quantiles should become thinner as the episodes progress, and box plots should converge to 100 % accuracy.

In *Few-shot user-definable radar-based hand gesture recognition at the edge* [88], the evaluation procedure is further elaborated to extract the underlying distribution of the generated box plots. The underlying density histograms of the first and last box plots for every experiment are compared with a Gaussian distribution. As the episodes progress, the histograms tend to move from an initial multimodal distribution (evaluation accuracy obtained for different complexities of tasks) to a negatively skewed distribution towards 100 % accuracy. The statistical comparison, also in terms of percentiles, highlights how the obtained accuracy histograms cannot be approximated by a Gaussian distribution. Another added feature represents the analysis of cumulative class-wise confusion matrices as episodes progress. Cumulative confusion matrices are generated over a set of episodes, appending to an array the class-wise predictions over tasks and analyzing them from a percentile perspective. Especially on test tasks, the more the main diagonal of the confusion matrix contains values closest to 100 as the episodes

progress, the more the model is actually generalizing to new scenarios.

In *Few-shot user-definable radar-based hand gesture recognition at the edge* [88], the same evaluation scenario is adapted to the regression scenario for the Conv-VAE training (Section 3.3). In such a case, the closer the intra-episode accuracy distribution is to zero, the better the estimation of test users will be. For breath estimation, a further evaluation is performed for test users by evaluating the error according to the average number of estimated breath peaks per 30-second session. This analysis has been performed on box plots built over equally distributed sets of sessions per range of detected breath peaks. In general, independently of the number of shots per experiment, the average and median losses are higher for less common cases of few (1-6) or many (10–14) peaks per 30s. Further, the whiskers and quartiles in such cases are much broader. By increasing the number of shots per experiment, the average and median values for edge cases decrease, as do the sizes of quartiles and whiskers. For the more common scenarios of 7 to 9 beats per session, the average error slightly increases. This is most likely due to more motion-corrupted sessions. Although the loss is defined to address such a problem, additional training examples may not contain enough information to improve the user's adaptation.

In *Context-Adaptable Radar-Based People Counting via Few-Shot Learning* [90], the developed evaluation framework is generalized over relation-based models. For the weighting network, the evaluation after every episode requires only a comparison between some support examples representative of the classes and the new test data. This is performed using both tasks sampled from the training and test meta-datasets. Since the prediction is performed by comparing support and test features via concatenation, there is no adaptation needed. This means that the evaluation can be done without the need to forget the learned evaluation weights, and it is much faster than in the optimization-based method. The adaptation procedure is the same for MAMW (Section 3.5), since in this case, the trained weighting network is just utilized for evaluation on samples from new tasks.

3.7. Exploration of active learning strategies for task fine-tuning

Active learning strategies such as pool-based sampling strategies are used in a wide range of tasks for traditional supervised learning to filter out the most useful data for training (Section 2.4). This is done by sampling some random data from an unlabeled data pool. The sampled instances are then evaluated to estimate the prediction uncertainty between classes. Then, an oracle, which may be the machine itself, chooses which data to label according to the chosen uncertainty criteria. The new labeled data are used in the next training steps as part of the training dataset. Normally, it is con-

sidered that the greater the uncertainty of an evaluated example, the more informative it is and thus the more useful it is for training.

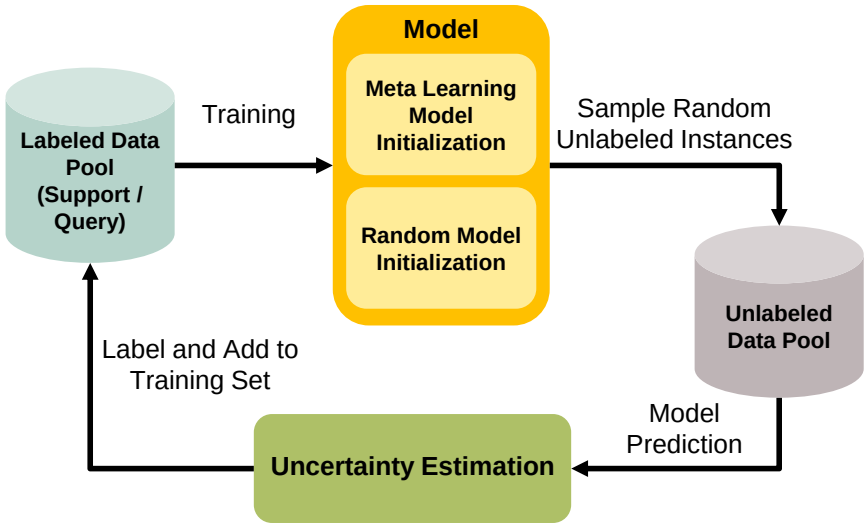


Figure 3.5: At each fine-tuning epoch of a task, the models are trained on all the available training data. Two different models are trained, one with random initialization and the other with episodic meta learning model initialization. The meta learning initialization is obtained through training in other contexts, thus making the model context-unaware of the fine-tuning scenario. The model prediction is done on examples randomly sampled from an unlabeled data pool. An estimate of uncertainty is made based on the predicted samples. Samples with higher uncertainty (and thus categorized as more significant) are added to the labeled data pool for the next training epoch.

The pool-based active learning strategy is explored in combination with meta learning in *Context-Adaptable Radar-Based People Counting via Few-Shot Learning* [90]. In this work, it is first of all proposed to leverage relation-based meta learning to obtain the best possible generalization on a set of defined tasks (Section 3.5). A pool-based active learning algorithm is then presented to fine-tune the Weighting-Injection Net (Section 3.4) on a new task by leveraging the acquired generalization knowledge during episodic training. A general depiction of the strategy is given in Figures. 3.5. The parameters learned in the episodic phase are used as model initialization for active learning. In a sequence of training epochs, the model learns how to solve the task from a training dataset (support and query) and estimates the prediction uncertainty on an evaluation set. That evaluation data are randomly sampled from a pool of unlabeled data. The examples with the highest uncertainty are then labeled by ground truth and added to the training set. At each step, support and query examples are also randomly sampled from the training

data pool. Per training epoch, the overall accuracy of a test set is also calculated. As epochs increase, the model effectively learns how to generalize over new data, labeling only those instances with the highest estimated information content. The approach is effective in the people counting task, leading to fine-tuning training in a new environment with an accuracy far greater than random model initialization. Compared with the model without prior generalization information as initialization, both the parameter sets learned with the relation-based approach and MAMW (Section 3.5) are significantly better. The classic relation-based training leads to about 30 % higher accuracy than random initialization over an average of three repetitions of the experiments. Further, the model employing random initialization collapses in some experiments to the value of random accuracy. This means that the random initialization fails to learn enough context information from the available data. Various possible uncertainty formulations that led to very similar fine-tuning performances have been taken into account in the active learning experiments.

3.8. Analysis of inference and adaptation trade-off of the generalization models

Inference time on single samples and adaptation time on a new task represent two very important KPIs for evaluating different meta learning strategies. As mentioned in Section 2.3, the various cutting-edge meta learning algorithms utilize different strategies of performance evaluation. The validation framework, as described in Section 3.6, allows for adequate evaluation of the various meta learning algorithms and topologies taken under consideration in this dissertation. Depending on the meta learning strategy, there might be a need for adaptation based on newly gathered data from a new context. Further, the size of the model can impact inference and lead to times that could be infeasible for real-time applications. In all papers that are part of the compendium in this dissertation, the trade-off between inference and adaptation time was evaluated, leveraging the developed evaluation framework.

In *One-shot meta learning for radar-based gesture sequences recognition* [87], a traditional CNN supervised training approach is compared with MAML episodic training on a meta-test sampled task. The traditional CNN is initialized to random weights after every task adaptation because a transfer learning approach fails to generalize to the new tasks after fine-tuning. On average, with 200 test samples, the generalization model obtained via meta learning requires only 8 support examples and roughly 1.5 seconds to reach the same performance that the traditional CNN obtains after 56 minutes and with 1,000 training samples.

In *Few-shot user-definable radar-based hand gesture recognition at the ed-*

ge [88], the adaptation time per task and inference time are measured for two different architectures. Specifically, these quantities are measured for an optimized architecture that uses part of a Conv-VAE as a backbone for feature extraction and for a pure CNN. On a quad-core CPU, the topology with the backbone leads to half the adaptation time compared to the CNN topology. The topology with a backbone in fact, is pre-trained on a set of training tasks and requires one order of magnitude fewer trainable parameters than the CNN. Both architectures, with an optimization-based training approach, are also compared with other state-of-the-art solutions. Both the feature extraction and CNN topologies are faster to adapt than the LGM-Net [105] but much slower than the weighting net [92]. For 5–way, anyway, the inference time needed by the presented solutions is less than all the state-of-the-art approaches considered. Even if the weighting net leads to better average test accuracy, the intrinsic topology requires a mapping between the multiple available support samples and the query. This specific characteristic makes the weighting net slower for inference with an increasing number of classes. At the edge, the trade-off between inference and adaptation time is analyzed for the computations on a pure ARM microcontroller and for the ARM microcontroller plus the Intel® Neural Compute Stick 2. The total time needed for adaptation becomes much more advantageous for several shots and several ways when the Compute Stick is connected, thanks to distributed computing capabilities and the short time needed to deploy binary models on the Compute Stick 2.

The various setups of the solution presented for hand gesture recognition in [88], are shown in Figure 3.6.

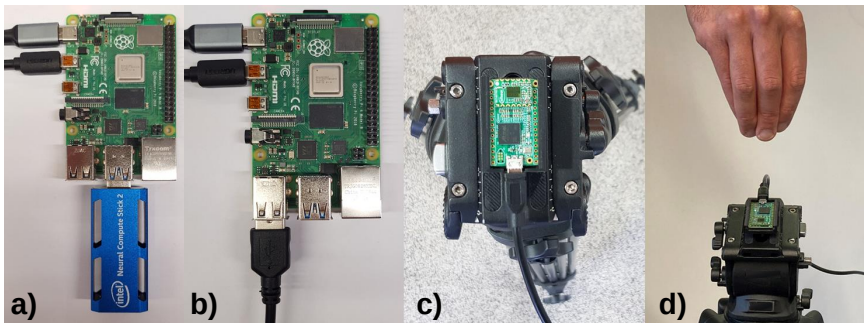


Figure 3.6: Data gathering scenarios and deployment of the optimization-based meta learning model at the Edge. In (a), the setup for inference is shown with Raspberry® Pi3 (ARM microprocessor) and the connected Intel® Neural Compute Stick 2. In (b), only the Raspberry® Pi3, used for data gathering. Shown in (c) is the Infineon XENSIV™ DEMO BGT60TR13C radar mounted on a tripod for data collection. An example of a hand gesture performed over the radar sensor is shown in (d).

In *Few-Shot User-Adaptable Radar-Based Breath Signal Sensing* [89], the task adaptation and inference time trade-off is computed over a regression problem with the formulated loss function (Section 3.4). The average adaptation time is computed over 300 test tasks of episodic adaptation with MAML. The episodic approach requires a steadily rising adaptation time as the number of shots grows, given four learning epochs per user adaptation (task). The adaptation time for 5– and 10– shot experiments is three to seven times greater than it is for a 1–shot. The 1–shot experiments may be thought of as providing the best trade-off because the mean loss does not drop significantly for 5- or 10-shots (less than 1 %). The number of shots utilized for user adaptation has no impact on the single inference time for MAML. In light of this, the inference time has been computed as a global average on all the 1–, 5–, and 10–shot experiments. As a result, a single inference time value of 4.30 ms has been estimated. The same task adaptation process as MAML for regression has been used to test other state-of-the-art algorithms. Only the generalization parameters in the intra-episodic stage are computed as a function of the algorithms. Algorithm-specific parameters are used in the evaluation phase, although first-order gradient optimization is always used in the adaptation. This means that although the parameter values change, the adaptation time is independent of the algorithm used. As a result, there is little to no difference in adaptation times across the selected methods with respect to MAML. The same holds true for the single sample inference time.

In *Context-Adaptable Radar-Based People Counting via Few-Shot Learning* [90], the computation of inference time is performed over relation-based models (Section 3.4). Despite requiring a custom topology for the classification of data pairs, relation-based models have a great advantage in task adaptation. In fact, after episodic adaptation and for any new task, the presented weighting-network-based solutions require only support examples to be used as a reference for the new classes. The prediction of the query class can then be done immediately afterwards, without any necessary adaptation time. The inference time, however, increases as the number of available training shots, i.e., the number of support examples per class, increases. For both MAMW (Section 3.5) and Weighting-Injection Net, the average inference time with a 1–shot is 14.46 ms on average and rises to 43.73 ms with 10 shots. This corresponds to an increase of about 67 % in single query inference time. For the state-of-the-art optimization-based algorithms under consideration, the average estimated inference over 10,000 examples is 33.47 ms, regardless of the number of shots. This is because there is no comparison between support and query but a direct mapping between examples and labels. The inference time of the state-of-the-art turns out to be better than the 10–shot relation-based experiments, at the expense, however, of accuracy about 15 % worse on the test data for the best-obtained models.

3.9. Exploration of meta learning implementations at the Edge

As outlined in Section 2.5, the deployment of meta learning solutions at the Edge can be expensive due to computation requirements for custom neural network topologies. Yet, the model agnosticism typical of optimization-based meta learning can ease deployment without setting important constraints on the model architecture and operations.

Few-shot user-definable radar-based hand gesture recognition at the edge [88] explores the implementation and adaptation of an optimization-based meta learning model at the Edge. The task adaptation is always performed on the Raspberry® Pi3 ARM microprocessor. After every task adaptation, the trainable parameters of the obtained model are in order, pruned, quantized, frozen, and then converted into a binary and an Extensible Markup Language (XML) file for being read by OpenVino for the Edge inference. The obtained binary file is transferred into the Intel® Neural Compute Stick 2. When connected to the Raspberry® Pi3, the Intel® Neural Compute Stick 2 is used for distributed computing on inference. With respect to the pure inference on the ARM Core, the single sample inference with connected Intel® Neural Compute Stick 2 takes around 5 ms rather than the 350 ms.

3.10. Projects Acknowledgments

The research developed during this doctoral program has been mostly funded by the funding European Projects **UpSim** and **ANDANTE**.

- The pan-European ITEA3 Project **UpSim** aims for Unleashing Potentials in Simulation by introducing quality management to modelling and simulation. Credible Digital Twins will be the game changer for accelerating innovation and reducing development costs in different industries. UpSim will boost virtual system development and will introduce collaboration processes that will ensure data availability in distributed development environments. Broad automation via continuous testing and AI supported simulation is focused in the combination with block-chain-based traceability and quality measures, which finally leads to credible Digital Twins for various applications – from smart engineering, virtual commissioning to predictive maintenance in system operation. This project has an initial duration of 36 months, formed by a consortium of 30 partners from 7 European Union countries. For the German partners, including Infineon Technologies AG, the project is funded by the German Federal Ministry of Education and Research.

  The research conducted in this doctoral program, related to the Up-

Sim project concerns the development of a radar system for in-cabin driver monitoring. For such a use case, it has been crucial to develop a pipeline for hand gesture recognition and vital parameter tracking. In this regard, the results are included in the following publications:

- Gianfranco Mauro, Mateusz Chmurski, Lorenzo Servadei, Manuel Pegalajar Cuellar, and Diego P. Morales-Santos. "Few-shot user-definable radar-based hand gesture recognition at the edge." IEEE Access 10: 29741-29759, 2022.
- Gianfranco Mauro, Maria De Carlos Diez, Julius Ott, Lorenzo Servadei, Manuel P. Cuellar, and Diego P. Morales-Santos. "Few-Shot User-Adaptable Radar-Based Breath Signal Sensing." Sensors 23, no. 2: 804, 2023.

- To create the AI foundations for future products in the edge Internet of Things (IoT) domain, the EU-funded ECSEL **ANDANTE** project aims to leverage innovative IC (integrated circuits) accelerators based on artificial and spiking neural networks in order to build strong hardware and software platforms for application developments. Moreover, the resulting IoT devices will combine extreme power efficiency with robust neuromorphic computing capabilities. By achieving efficient cross-fertilization between major European foundries, chip designers, system houses, application companies and research partners, the project will build and expand the European ecosystem around the definition, development, production and application of neuromorphic ICs. The project's work will promote innovative hardware and software deep-learning solutions for future IoT at the edge products that combine extreme power efficiency as well as robust and powerful cognitive computing capabilities. This project has an initial duration of 36 months, formed by a consortium of 30 partners from 7 European Union countries. For the German partners, including Infineon Technologies AG, the project is funded by the German Federal Ministry of Education and Research.

  The related doctoral program research for the ANDANTE project, yield the development of an indoor people counting and tracking solution using radar. The main objective has been to improve the adaptability of the system in new environments and conditions compared to the state-of-the-art. In this regard, the results are included in the following publication:

  - Gianfranco Mauro, Ignacio Martinez-Rodriguez, Julius Ott, Lorenzo Servadei, Robert Wille, Manuel P. Cuellar, and Diego P. Morales-Santos. "Context-Adaptable Radar-Based People Counting via Few-Shot Learning." Applied Intelligence, 1-29, 2023.

3.11. Collaborations

Apart from the paper compendium, several co-authored publications have been the result of collaborative projects with research and industrial partners. Specifically, collaboration and co-authorship contributions resulted in the following papers:

1. Mateusz Chmurski, Gianfranco Mauro, Avik Santra, Mariusz Zubert and Gökberk Dagasan. Highly-optimized radar-based gesture recognition system with depthwise expansion module. Sensors, 21(21), 7298, 2021.

2. Julius Ott, Lorenzo Servadei, Gianfranco Mauro, Thomas Stadelmayer, Avik Santra and Robert Wille. Uncertainty-based Meta-Reinforcement Learning for Robust Radar Tracking. In 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 1476-1483). IEEE, 2022.

3. Jakob Valtl, Javier Mendez-Gomez, Gianfranco Mauro, Antonio Cabrera and Vadim Issakov. Investigation for the need of traditional data-preprocessing when applying artificial neural networks to FMCW-radar data. In 2022 29th International Conference on Systems, Signals and Image Processing (IWSSIP) (pp. 1-4). IEEE, 2022.

4. Borja Saez-Mingorance, Javier Mendez-Gomez, Gianfranco Mauro, Encarnacion Castillo-Morales, Manuel Pegalajar-Cuellar and Diego P Morales-Santos. Air-writing character recognition with ultrasonic transceivers. Sensors, 21(20), 6700, 2021.

5. Julius Ott, Lorenzo Servadei, Jose Arjona-Medina, Enrico Rinaldi, Gianfranco Mauro and Daniela Lopera Sánchez, Michael Stephan, Thomas Stadelmayer, Avik Santra and Robert Wille. MEET: A Monte Carlo Exploration-Exploitation Trade-Off for Buffer Sampling. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1-5). IEEE, 2023.

6. Muhammad Arsalan, Avik Santra, Mateusz Chmurski, Moamen El-Masry, Gianfranco Mauro and Vadim Issakov. Radar-based gesture recognition system using spiking neural network. In 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA) (pp. 1-5). IEEE, 2021.

Chapter 4

Conclusions

One never notices what has been done;
one can only see what remains to be done.

Marie Curie

This doctoral research has focused on the design and optimization of context generalization techniques in radar applications by leveraging only the few available data instances. The adopted approach has been exploratory, researching and adapting few-shot approaches and meta learning to three distinct radar-based use cases. The use of active learning as a fine-tuning method for pre-trained meta learning models has also been researched. For each defined use case, the main pipeline steps have been designed and optimized for the episodic learning approach. For the generation of a proof-of-concept solution, these steps mainly involved the collection of a dataset, extraction of useful information, generation of learning tasks, design of an ML model, training and validation of the ML model, and finally model deployment at the Edge. The main conclusions and contributions of this doctoral research are listed as follows:

- The first important contribution of this research has been harnessing the advantages of radar technology for context generalization. The use of mm-wave FMCW radar technology for this research (Infineon XENSIV™ DEMO BGT60TR13C), has been based on extensive investigation covering both the technology itself and potential applications. The chosen radar-based use cases, namely hand gesture recognition, breath signal sensing, and in-door people counting, have been carefully selected to explore information processing complexities at various sensing magnitudes. In all four papers that are part of the compendium [87, 88, 89, 90], the research has addressed the need for generalization in new scenarios, considering factors such as user characteristics, en-

vironmental variations, and functional parameters of the devices. The research contributions for objectives 1.a and 1.b of Section 1.2, included specific descriptions of use cases, strategies for generating tasks, and the creation of meta-datasets for each application. This yields a better evaluation of the context generalization approach. The results detailed in the compendium publications demonstrate how the proposed methods can effectively achieve context adaptability in radar-based use cases.

In summary, this research has made significant strides in harnessing radar technology for context generalization, paving the way for more scalable, adaptable, and privacy-compliant solutions in various real-world use cases. The presented methodologies and findings have the potential to drive further advancements in radar-based systems and contribute to the broader field of sensor applications and context learning.

- Another relevant focus of the research has been tailoring radar data processing for the episodic learning approach. Mainly the Objectives 2.a and 2.b. of Section 1.2. For context generalization, non-recurrent data handling techniques have been used. In all the attached publications [87, 88, 89, 90], different data processing techniques have been adopted and evaluated via metrics such as the number of trainable parameters and task performance. For all the relevant use cases, the gathered time information has been mapped into channel representations as part of the data preprocessing. The use of machine learning, particularly CNNs and autoencoders, facilitated feature extraction from radar data, which inherently lack of interpretability. Especially, the use of Conv-VAEs played a crucial role in reducing data dimensionality while retaining important information. The model with a Conv-VAE-based backbone, enabled in [88], efficient few-shot adaptation and edge deployment with minimal loss in accuracy. Together with the Conv-VAE, the custom loss function designed for the task of breath signal sensing in [89] led to a user-adaptable radar-based sensing solution with robustness to motion corruption. The designed topologies, such as the Weighting-Injection Net in [90], helped to robustly achieve context adaptation thanks to the inherent instance comparison strategy and feature extraction.

  The findings provide valuable insights for the design and implementation of radar-based solutions with efficient data processing and adaptable learning capabilities.

- An extensive part of the research has been devoted to the research, optimization, and standardized evaluation of meta learning techniques (Objectives 3.a, 3.b, and 3.c of Section 1.2). A special focus has been

given to optimization-based and relation-based algorithms.

The model-agnosticism property of optimization-based methods, enabled decoupling data handling and processing requirements from the chosen training algorithms in [87, 88, 89]. In [88], specific strategies have been designed and evaluated to enhance the stability and performance of optimization-based algorithms for hand gesture recognition. As highlighted in the achievements 3.5, these approaches showcased different performances depending on the experiment setup and network topology. In [89], the optimization-based algorithms have been evaluated on a regression task for a non-contact estimation of the breath signal from radar data. Additionally, relation-based algorithms have been investigated, as they enable powerful context generalization by learning to classify matching characteristics of instances. In [90], a novel algorithm called Model-agnostic meta-weighting (MAMW) has been introduced to combine the advantages of optimization-based and relation-based meta learning. MAMW aimed to increase stability in few-shot experiments, particularly for people counting tasks, using the Weighting-Injection Net as the topology.

Through the state-of-the-art investigation, a lack of a unified evaluation framework for meta learning algorithms has been observed. As a result, a new standardized evaluation framework has been developed to facilitate comparison and analysis of different optimization-based and relation-based approaches. The evaluation framework (Section 3.6), facilitates the comparison of various algorithms by providing numerical and graphical evaluations in a consistent format. The generalization performance of models is assessed after each episodic adaptation, for both training and test tasks. Additionally, the framework assesses the trade-off between single-sample inference time and training time to adapt to a new task. The initial evaluation for optimization-based algorithms included analysis of box plots and cumulative class-wise confusion matrices for classification tasks [87, 88]. The framework has been further extended to regression tasks in [89] and to relation-based algorithms in [90]. The conducted analysis in the various use cases demonstrated that the designed evaluation approach can effectively reveal and compare the generalization capabilities of different models. The same framework has also been employed for assessing the generalization capabilities of models trained on the public dataset Omniglot in [90].

In summary, the conducted research significantly contributes to the optimization and evaluation of optimization-based and relation-based meta learning algorithms. The main outcome has been the development of new strategies and algorithms to improve learning stability and performance. Aside from that, the development of a comprehensive eva-

luation framework enabled the accurate comparison of the developed techniques to the state-of-the-art.

- Part of the research conducted explored the use of active learning strategies on top of pre-trained generalization models to optimize model performance with limited labeled instances (Objectives 3.d of Section 1.2). One of the main objectives in [90], has been to investigate whether episodic meta learning, which leverages prior experience from similar tasks, could benefit active learning fine-tuning. The research highlighted that, on top of an application-aware model, active learning effectively filters out relevant data from an unlabeled pool, leading to improved model performance with respect to random initialization. Specifically, the use of pool-based sampling strategies allowed the selection of only informative examples via prediction uncertainty analysis, contributing to better training and adaptation to new contexts.

  Overall, the findings for the radar-based in-door people counting task highlight an important synergy between active learning and meta learning. The use of previously acquired information can be used not only to simplify adaptation in new contexts but also to filter out informative data through active learning. Further exploration and refinements of the integration between meta learning and active learning can have important future implications beyond radar-based use cases.

- To assess the actual impact of the episodic learning approach on potential deployed solutions, part of the research has been devoted to the trade-off between adaptation and inference (Objectives 4.a and 4.b of Section 1.2). The time trade-off between inference on a single instance and task adaptation has been analyzed in all the papers that are part of the compendium [87, 88, 89, 90]. For this investigation, the developed evaluation framework (Section 3.6) has been essential in standardizing the evaluation and comparison of different meta learning algorithms. The evaluation performed in all the researched use cases highlighted the advantages of meta learning over traditional supervised training approaches for adaptation. In [87], the meta learning-based model achieved comparable performance to a CNN trained with the classic supervised approach. Thanks to the prior knowledge acquired in other contexts, the meta learning-based model required a significantly smaller number of support examples and a shorter adaptation time than the CNN, which required extensive training data and time. In [88], further analysis has been done at the Edge, to explore the limitations and deployment capabilities of meta learning algorithms given limited computation. Despite the computation and model size challenges for meta learning, the agnostic nature of optimization-based algorithms simplified the Edge deployment on the Intel® Neural Com-

pute Stick 2 and on the Raspberry® Pi3 ARM microprocessor. Even with computation constraints at the Edge, the agnosticism property of optimization-based meta learning proved to be valuable in enabling deployment without placing topological constraints on the model.

In summary, the conducted research contributes to the advancement of deploying meta learning solutions at the Edge. The findings, also in terms of adaptation and inference trade-offs, offer valuable insights into the feasibility of deploying such solutions in real-world applications. Further exploration of techniques such as model pruning and quantization tuned for meta learning can foster the development of efficient and context-adaptable ML models at the Edge.

- All remarkable results obtained during this doctoral program have been published in scientific journals and presented at a conference, contributing to the state-of-the-art. Such papers represent the compendium of publications attached to the thesis in the next chapters. The conference paper has been published with Springer for the *31st International Conference on Artificial Neural Networks* (ICANN). The scientific journals have been *Springer Nature Applied Intelligence*, *MDPI Sensors* and *IEEE Access*.

According to the research carried out, a considerable enhancement of the self-learning capability of models can be expected in the near future for sensor applications. Approaches suitable for extracting and comparing useful information from data will strengthen the generalization features of artificial intelligence. Most of the research will probably leverage techniques suitable for context generalization, such as meta learning and active learning, given their emerging importance in a wide variety of applications, such as industrial and medical. The computational development of Edge devices and on-chip systems will enable real-time context generalization for a large variety of applications. This will allow sensor-based solutions to adapt robustly, with little data, and in a short time, to new scenarios and users. Many applications, such as autonomous driving, will benefit from such algorithmic improvements.

4.1. Future trends

Based on the experience gained through the doctoral research, some main trends in the field of sensing applications can be foreseen. This could be the object of further research.

- **Emerging meta learning strategies:** in recent years, AI research has focused on Implicit Neural Representation (INR). In this regard, meta learning can be extremely useful for context and sparse data

generalization [106]. The idea behind INRs is to represent signals as continuous functions parameterized by a neural network that maps the domain to the codomain, e.g., mapping the coordinates of an image to their pixel values. One of the main advantages of INRs is that they no longer constrain data to their spatial representation but parameterize them. This can overcome the problem of handling discrete data with limited resolution, such as audio signals or grids of pixels. Once the implicit representation of the data is learned, INRs can also be used as generational models, even in 3-D [107]. In relation to this, meta learning can be used to generalize on rarer but important representations based on prior information obtained from available data. In this regard, architectures capable of automatically and efficiently extracting information from available data, such as Transformers, are beginning to be used as meta learners [108].

The forecast is that meta learning will be heavily employed for data reconstruction and generation, especially through INRs. This is because information acquired from other contexts can be effectively employed to learn parametric representations of the data that do not overfit individual samples. This could be an important part of the next generation of image and video AI generators.

- **Sensor applications and context learning:** in many industrial case studies nowadays, it is essential to make real-time decisions using information from a large number of sensors and a large amount of data. A prime example is autonomous driving, where passenger safety may depend on decisions made in split-second moments. Normally, information collected from various sensors is processed, fused, and processed in part by ML algorithms to speed up decisions [109, 110]. In particular parts of the world, such as the United States, self-driving Taxi solutions already exist. While efficient, such systems have been trained largely on data collected in driving and traffic scenarios typical of the geographic area. This means that in other areas of the world with different driving and road conditions, such solutions may not be viable. Rather than training the models from scratch in each geographic area, information acquired in other scenarios can be effectively employed in a meta-learning approach to enable faster learning and decision-making in new contexts. In this way, all sensor applications could leverage a much larger amount of available information. This can be further enhanced by fusion approaches between multiple units of a sensor or types of sensors, such as radar, LiDAR, and cameras.

  The prediction is that, in the short future, it will be possible to generalize to uncommon contexts even while having access to a small amount of data, thanks to meta learning. This may have great implications for

society, thanks to autonomous driving, for example.

- **Edge deployment of context-adaptable solutions:** While it will be possible in the near future, to develop large-scale context-adaptable solutions, it will also be necessary to have powerful and power-efficient Edge devices capable of processing information and learning in real-time.

  Such solutions will also need custom methodologies for pruning and quantizing model parameters so that generalization information is not lost in the deployment phase. This means that state-of-the-art hardware and potentially neuromorphic computing [111], will be needed to enable such applications through low inference and scalable production in the market. On the other hand, specific model compression techniques, such as pruning and quantization, will have to be well adapted to meta learning to avoid the loss of valuable generalization information during deployment. Algorithms that pose meta learning goals to avoid task overfitting and pruning of channels in very deep neural networks are already being researched [112, 113].

  The forecast is that, thanks to the advent of new Edge technologies and model size reduction techniques geared toward generalization, there will be AI solutions with enormous and robust adaptive capabilities in the near future. This will make it possible and fast to deploy complex solutions such as autonomous driving in new contexts while also enabling continuous system evolution and adaptation from newly available data.

References

[1] Mohd Javaid, Abid Haleem, Ravi Pratap Singh, Shanay Rab, and Rajiv Suman. Significance of sensors for industry 4.0: Roles, capabilities, and applications. *Sensors International*, 2:100110, 2021.

[2] S Joe Qin. Survey on data-driven industrial process monitoring and diagnosis. *Annual reviews in control*, 36(2):220–234, 2012.

[3] Rashmi Saini and Vinod Maan. Human activity and gesture recognition: a review. In *2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)*, pages 1–2. IEEE, 2020.

[4] Oscar D Lara and Miguel A Labrador. A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials*, 15(3):1192–1209, 2012.

[5] Zhihan Lv and Yuxi Li. Wearable sensors for vital signs measurement: A survey. *Journal of Sensor and Actuator Networks*, 11(1):19, 2022.

[6] L Minh Dang, Kyungbok Min, Hanxiang Wang, Md Jalil Piran, Cheol Hee Lee, and Hyeonjoon Moon. Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognition*, 108:107561, 2020.

[7] Charmi Jobanputra, Jatna Bavishi, and Nishant Doshi. Human activity recognition: A survey. *Procedia Computer Science*, 155:698–703, 2019.

[8] Fatai Idowu Sadiq, Ali Selamat, and Roliana Ibrahim. Human activity recognition prediction for crowd disaster mitigation. In *Asian Conference on Intelligent Information and Database Systems*, pages 200–210. Springer, 2015.

[9] Gheorghe Sebestyen, Ionut Stoica, and Anca Hangan. Human activity recognition and monitoring for elderly people. In *2016 IEEE 12th international conference on intelligent computer communication and processing (ICCP)*, pages 341–347. IEEE, 2016.

[10] Franco Cicirelli, Giancarlo Fortino, Andrea Giordano, Antonio Guerrieri, Giandomenico Spezzano, and Andrea Vinci. On the design of smart homes: A framework for activity recognition in home environment. *Journal of medical systems*, 40(9):1–17, 2016.

[11] Siddharth S Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial intelligence review*, 43(1):1–54, 2015.

[12] Sushmita Mitra and Tinku Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(3):311–324, 2007.

[13] Duarte Dias and João Paulo Silva Cunha. Wearable health devices—vital sign monitoring, systems and technologies. *Sensors*, 18(8):2414, 2018.

[14] Zhihua Wang, Zhaochu Yang, and Tao Dong. A review of wearable technologies for elderly care that can accurately track indoor position, recognize physical activities and monitor vital signs in real time. *Sensors*, 17(2):341, 2017.

[15] Andrea Prati, Caifeng Shan, and Kevin I-Kai Wang. Sensors, vision and networks: From video surveillance to activity recognition and health monitoring. *Journal of Ambient Intelligence and Smart Environments*, 11(1):5–22, 2019.

[16] Konstantinos A Tarabanis, Peter K Allen, and Roger Y Tsai. A survey of sensor planning in computer vision. *IEEE transactions on Robotics and Automation*, 11(1):86–104, 1995.

[17] Mazin Alshamrani. Iot and artificial intelligence implementations for remote healthcare monitoring systems: A survey. *Journal of King Saud University-Computer and Information Sciences*, 34(8):4687–4701, 2022.

[18] Dragos Mocrii, Yuxiang Chen, and Petr Musilek. Iot-based smart homes: A review of system architecture, software, communications, privacy and security. *Internet of Things*, 1:81–98, 2018.

[19] Prerit Datta, Akbar Siami Namin, and Moitrayee Chatterjee. A survey of privacy concerns in wearable devices. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4549–4553. IEEE, 2018.

[20] Zhe Geng, He Yan, Jindong Zhang, and Daiyin Zhu. Deep-learning for radar: A survey. *IEEE Access*, 9:141800–141818, 2021.

[21] Andrea Wrabel, Roland Graef, and Tobias Brosch. A survey of artificial intelligence approaches for target surveillance with radar sensors. *IEEE Aerospace and Electronic Systems Magazine*, 36(7):26–43, 2021.

[22] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern recognition letters*, 119:3–11, 2019.

[23] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

[24] Xiaoxu Li, Zhuo Sun, Jing-Hao Xue, and Zhanyu Ma. A concise review of recent few-shot meta-learning methods. *Neurocomputing*, 456:463–468, 2021.

[25] Abdullatif Köksal, Timo Schick, and Hinrich Schütze. Meal: Stable and active learning for few-shot prompting. *arXiv preprint arXiv:2211.08358*, 2022.

[26] Joaquin Vanschoren. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548*, 2018.

[27] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

[28] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[29] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM Computing Surveys (CSUR)*, 54(9):1–40, 2021.

[30] Djamila Romaissa Beddiar, Brahim Nini, Mohammad Sabokrou, and Abdenour Hadid. Vision-based human activity recognition: a survey. *Multimedia Tools and Applications*, 79:30509–30555, 2020.

[31] Enida Cero Dinarević, Jasmina Baraković Husić, and Sabina Barakovi ć. Issues of human activity recognition in healthcare. In *2019 18th international symposium infoteh-jahorina (infoteh)*, pages 1–6. IEEE, 2019.

[32] Ruchika Sinhal, Kavita Singh, and Anuraj Shankar. Estimating vital signs through non-contact video-based approaches: A survey. In *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*, pages 139–141. IEEE, 2017.

[33] Jianfei Yang, Yuecong Xu, Haozhi Cao, Han Zou, and Lihua Xie. Deep learning and transfer learning for device-free human activity recognition: A survey. *Journal of Automation and Intelligence*, 1(1):100007, 2022.

[34] Adnan Farooq and Chee Sun Won. A survey of human action recognition approaches that use an rgb-d sensor. *IEIE Transactions on Smart Processing and Computing*, 4(4):281–290, 2015.

[35] Daniel McDuff. Camera measurement of physiological vital signs. *ACM Computing Surveys*, 55(9):1–40, 2023.

[36] Harshit Mittal and Neeraj Garg. A survey on automatic vehicles using computer vision. *Available at SSRN 4387778*, 2023.

[37] Irvan B Arief-Ang, Flora D Salim, and Margaret Hamilton. Cd-hoc: indoor human occupancy counting using carbon dioxide sensor data. *arXiv preprint arXiv:1706.05286*, 2017.

[38] Uday Kamal, Shamir Ahmed, Tarik Reza Toha, Nafisa Islam, and ABM Alim Al Islam. Intelligent human counting through environmental sensing in closed indoor settings. *Mobile Networks and Applications*, 25:474–490, 2020.

[39] Sizhen Bian, Mengxi Liu, Bo Zhou, and Paul Lukowicz. The state-of-the-art sensing techniques in human activity recognition: A survey. *Sensors*, 22(12):4596, 2022.

[40] Hefeng Wu, Chengying Gao, Yirui Cui, and Ruomei Wang. Multipoint infrared laser-based detection and tracking for people counting. *Neural Computing and Applications*, 29:1405–1416, 2018.

[41] Rikke Gade and Thomas B Moeslund. Thermal cameras and applications: a survey. *Machine vision and applications*, 25:245–262, 2014.

[42] EFJ Ring and Kurt Ammer. Infrared thermal imaging in medicine. *Physiological measurement*, 33(3):R33, 2012.

[43] Mahmudul Hasan, Junichi Hanawa, Riku Goto, Ryota Suzuki, Hisato Fukuda, Yoshinori Kuno, and Yoshinori Kobayashi. Lidar-based detection, tracking, and property estimation: A contemporary review. *Neurocomputing*, 2022.

[44] Csaba Benedek, Bence Gálai, Balázs Nagy, and Zsolt Jankó. Lidar-based gait analysis and activity recognition in a 4d surveillance system. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(1):101–113, 2016.

[45] Bruno Rodrigues, Lukas Müller, Eder J Scheid, Muriel F Franco, Christian Killer, and Burkhard Stiller. Laflector: A privacy-preserving lidar-based approach for accurate indoor tracking. In *2021 IEEE 46th Conference on Local Computer Networks (LCN)*, pages 367–370. IEEE, 2021.

[46] Md Siddat Bin Nesar, Karis Trippe, Ryan Stapley, Bradley M Whitaker, and Bryce Hill. Improving touchless respiratory monitoring via lidar orientation and thermal imaging. In *2022 IEEE Aerospace Conference (AERO)*, pages 1–8. IEEE, 2022.

[47] Gayatri Chittimoju and Usha Devi Yalavarthi. A comprehensive review on millimeter waves applications and antennas. In *Journal of Physics: Conference Series*, volume 1804, page 012205. IOP Publishing, 2021.

[48] Lukas Piotrowsky, Timo Jaeschke, Simon Kueppers, Jan Siska, and Nils Pohl. Enabling high accuracy distance measurements with fmcw radar sensors. *IEEE Transactions on Microwave Theory and Techniques*, 67(12):5360–5371, 2019.

[49] Arthur Venon, Yohan Dupuis, Pascal Vasseur, and Pierre Merriaux. Millimeter wave fmcw radars for perception, recognition and localization in automotive applications: A survey. *IEEE Transactions on Intelligent Vehicles*, 7(3):533–555, 2022.

[50] J Le Kernec, F Fioranelli, C Ding, H Zhao, L Sun, H Hong, O Romain, and J Lorandel. Radar sensing in assisted living: An overview. In *2019 IEEE MTT-S International Microwave Biomedical Conference (IMBioC)*, volume 1, pages 1–4. IEEE, 2019.

[51] Fahad Jibrin Abdu, Yixiong Zhang, Maozhong Fu, Yuhan Li, and Zhenmiao Deng. Application of deep learning on millimeter-wave radar signals: A review. *Sensors*, 21(6):1951, 2021.

[52] Xinyu Li, Yuan He, and Xiaojun Jing. A survey of deep learning-based human activity recognition in radar. *Remote Sensing*, 11(9):1068, 2019.

[53] Shahzad Ahmed, Karam Dad Kallu, Sarfaraz Ahmed, and Sung Ho Cho. Hand gestures recognition using radar sensors for human-computer-interaction: A review. *Remote Sensing*, 13(3):527, 2021.

[54] Peijun Zhao, Chris Xiaoxuan Lu, Jianan Wang, Changhao Chen, Wei Wang, Niki Trigoni, and Andrew Markham. mid: Tracking and identifying people with millimeter wave radar. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 33–40. IEEE, 2019.

[55] Victor C Chen. *The micro-Doppler effect in radar*. Artech house, 2019.

[56] Ameen Bin Obadi, Ping Jack Soh, Omar Aldayel, Muataz Hameed Al-Doori, Marco Mercuri, and Dominique Schreurs. A survey on vital signs detection using radar techniques and processing with fpga implementation. *IEEE Circuits and Systems Magazine*, 21(1):41–74, 2021.

[57] Antonio Iula. Ultrasound systems for biometric recognition. *Sensors*, 19(10):2317, 2019.

[58] Zhengjie Wang, Yushan Hou, Kangkang Jiang, Wenwen Dou, Chengming Zhang, Zehua Huang, and Yinjing Guo. Hand gesture recognition based on active ultrasonic sensing of smartphone: a survey. *IEEE Access*, 7:111897–111922, 2019.

[59] Zhengjie Wang, Yushan Hou, Kangkang Jiang, Chengming Zhang, Wenwen Dou, Zehua Huang, and Yinjing Guo. A survey on human behavior recognition using smartphone-based ultrasonic signal. *IEEE Access*, 7:100581–100604, 2019.

[60] Francis A Duck, Andrew Charles Baker, and Hazel C Starritt. *Ultrasound in medicine*. CRC Press, 2020.

[61] Se Dong Min, Jin Kwon Kim, Hang Sik Shin, Yong Hyeon Yun, Chung Keun Lee, and Myoungho Lee. Noncontact respiration rate measurement system using an ultrasonic proximity sensor. *IEEE sensors journal*, 10(11):1732–1739, 2010.

[62] Michele Ambrosanio, Stefano Franceschini, Giuseppe Grassini, and Fabio Baselice. A multi-channel ultrasound system for non-contact heart rate monitoring. *IEEE Sensors Journal*, 20(4):2064–2074, 2019.

[63] Subhas Chandra Mukhopadhyay. Wearable sensors for human activity monitoring: A review. *IEEE sensors journal*, 15(3):1321–1330, 2014.

[64] Shwetank Dattatraya Mamdiwar, Zainab Shakruwala, Utkarsh Chadha, Kathiravan Srinivasan, and Chuan-Yu Chang. Recent advances on iot-assisted wearable sensor systems for healthcare monitoring. *Biosensors*, 11(10):372, 2021.

[65] Niall Lyons, Avik Santra, and Ashutosh Pandey. Improved deep representation learning for human activity recognition using imu sensors. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 326–332. IEEE, 2021.

[66] Shuo Jiang, Peiqi Kang, Xinyu Song, Benny PL Lo, and Peter B Shull. Emerging wearable interfaces and algorithms for hand gesture recognition: A survey. *IEEE Reviews in Biomedical Engineering*, 15:85–102, 2021.

[67] Sumit Majumder, Tapas Mondal, and M Jamal Deen. Wearable sensors for remote health monitoring. *Sensors*, 17(1):130, 2017.

[68] Wei Wang, Alex X Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. Device-free human activity recognition using commercial wifi devices. *IEEE Journal on Selected Areas in Communications*, 35(5):1118–1131, 2017.

[69] Daqing Zhang, Youwei Zeng, Fusang Zhang, and Jie Xiong. Wifi csi-based vital signs monitoring. In *Contactless Vital Signs Monitoring*, pages 231–255. Elsevier, 2022.

[70] Eric Mason, Bariscan Yonel, and Birsen Yazici. Deep learning for radar. In *2017 IEEE Radar Conference (RadarConf)*, pages 1703–1708. IEEE, 2017.

[71] Jakob Valtl, Javier Mendez, Gianfranco Mauro, Antonio Cabrera, and Vadim Issakov. Investigation for the need of traditional data-preprocessing when applying artificial neural networks to fmcw-radar data. In *2022 29th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 1–4. IEEE, 2022.

[72] Eiji Hayashi, Jaime Lien, Nicholas Gillian, Leonardo Giusti, Dave Weber, Jin Yamanaka, Lauren Bedal, and Ivan Poupyrev. Radarnet: Efficient gesture recognition technique utilizing a miniature radar sensor. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2021.

[73] Umer Saeed, Syed Yaseen Shah, Syed Aziz Shah, Jawad Ahmad, Abdullah Alhumaidi Alotaibi, Turke Althobaiti, Naeem Ramzan, Akram Alomainy, and Qammer H Abbasi. Discrete human activity recognition and fall detection by combining fmcw radar data of heterogeneous environments for independent assistive living. *Electronics*, 10(18):2237, 2021.

[74] Michael Stephan, Souvik Hazra, Avik Santra, Robert Weigel, and Georg Fischer. People counting solution using an fmcw radar with knowledge distillation from camera data. In *2021 IEEE Sensors*, pages 1–4. IEEE, 2021.

[75] Mateusz Chmurski, Mariusz Zubert, Kay Bierzynski, and Avik Santra. Analysis of edge-optimized deep learning classifiers for radar-based gesture recognition. *IEEE Access*, 9:74406–74421, 2021.

[76] Zhenyuan Zhang, Zengshan Tian, and Mu Zhou. Latern: Dynamic continuous hand gesture recognition using fmcw radar sensor. *IEEE Sensors Journal*, 18(8):3278–3289, 2018.

[77] Jae-Woo Choi, Si-Jung Ryu, and Jong-Hwan Kim. Short-range radar based real-time hand gesture recognition using lstm encoder. *IEEE Access*, 7:33610–33618, 2019.

[78] A Helen Victoria and G Maragatham. Activity recognition of fmcw radar human signatures using tower convolutional neural networks. *Wireless Networks*, pages 1–17, 2021.

[79] Lorenzo Servadei, Huawei Sun, Julius Ott, Michael Stephan, Souvik Hazra, Thomas Stadelmayer, Daniela Sanchéz Lopera, Robert Wille, and Avik Santra. Label-aware ranked loss for robust people counting using automotive in-cabin radar. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3883–3887. IEEE, 2022.

[80] Jian Gong, Xinyu Zhang, Kaixin Lin, Ju Ren, Yaoxue Zhang, and Wenxun Qiu. Rf vital sign sensing under free body movement. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(3):1–22, 2021.

[81] Dingyang Wang, Sungwon Yoo, and Sung Ho Cho. Experimental comparison of ir-uwb radar and fmcw radar for vital signs. *Sensors*, 20(22):6695, 2020.

[82] Gor Hakobyan and Bin Yang. High-performance automotive radar: A review of signal processing algorithms and modulation schemes. *IEEE Signal Processing Magazine*, 36(5):32–44, 2019.

[83] Hui-Shyong Yeo and Aaron Quigley. Radar sensing in human-computer interaction. *interactions*, 25(1):70–73, 2017.

[84] Mamady Kebe, Rida Gadhafi, Baker Mohammad, Mihai Sanduleanu, Hani Saleh, and Mahmoud Al-Qutayri. Human vital signs detection methods and potential using radars: A review. *Sensors*, 20(5):1454, 2020.

[85] Sylvia T Kouyoumdjieva, Peter Danielis, and Gunnar Karlsson. Survey of non-image-based approaches for counting people. *IEEE Communications Surveys & Tutorials*, 22(2):1305–1336, 2019.

[86] Mostafa Alizadeh, George Shaker, João Carlos Martins De Almeida, Plinio Pelegrini Morita, and Safeddin Safavi-Naeini. Remote monitoring of human vital signs using mm-wave fmcw radar. *IEEE Access*, 7:54958–54968, 2019.

[87] Gianfranco Mauro, Mateusz Chmurski, Muhammad Arsalan, Mariusz Zubert, and Vadim Issakov. One-shot meta-learning for radar-based gesture sequences recognition. In *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part II 30*, pages 500–511. Springer, 2021.

[88] Gianfranco Mauro, Mateusz Chmurski, Lorenzo Servadei, Manuel Pegalajar-Cuellar, and Diego P Morales-Santos. Few-shot user-definable radar-based hand gesture recognition at the edge. *IEEE Access*, 10:29741–29759, 2022.

[89] Gianfranco Mauro, Maria De Carlos Diez, Julius Ott, Lorenzo Servadei, Manuel P Cuellar, and Diego P Morales-Santos. Few-shot user-adaptable radar-based breath signal sensing. *Sensors*, 23(2):804, 2023.

[90] Gianfranco Mauro, Ignacio Martinez-Rodriguez, Julius Ott, Lorenzo Servadei, Robert Wille, Manuel P Cuellar, and Diego P Morales-Santos. Context-adaptable radar-based people counting via few-shot learning. *Applied Intelligence*, HH(HH):HH, 2023.

[91] Zhongyu Fan, Haifeng Zheng, and Xinxin Feng. A meta-learning-based approach for hand gesture recognition using fmcw radar. In *2020 International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 522–527. IEEE, 2020.

[92] Xianglong Zeng, Chaoyang Wu, and Wen-Bin Ye. User-definable dynamic hand gesture recognition based on doppler radar and few-shot learning. *IEEE Sensors Journal*, 21(20):23224–23233, 2021.

[93] Huawei Hou, Suzhi Bi, Lili Zheng, Xiaohui Lin, Yuan Wu, and Zhi Quan. Dasecount: Domain-agnostic sample-efficient wireless indoor crowd counting via few-shot learning. *IEEE Internet of Things Journal*, 2022.

[94] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[95] Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

[96] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

[97] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. The omniglot challenge: a 3-year progress report. *Current Opinion in Behavioral Sciences*, 29:97–104, 2019.

[98] Punit Kumar and Atul Gupta. Active learning query strategies for classification, regression, and clustering: a survey. *Journal of Computer Science and Technology*, 35:913–945, 2020.

[99] Lucas Baier, Fabian Jöhren, and Stefan Seebacher. Challenges in the deployment and operation of machine learning in practice. In *ECIS*, volume 1, 2019.

[100] Yong Wang, Wen Wang, Mu Zhou, Aihu Ren, and Zengshan Tian. Remote monitoring of human vital signs based on 77-ghz mm-wave fmcw radar. *Sensors*, 20(10):2999, 2020.

[101] Saverio Trotta, Dave Weber, Reinhard W Jungmaier, Ashutosh Baheti, Jaime Lien, Dennis Noppeney, Maryam Tabesh, Christoph Rumpler, Michael Aichner, Siegfried Albel, et al. Soli: A tiny device for a new human machine interface. In *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, volume 64, pages 42–44. IEEE, 2021.

[102] Raymond J Weber and Yikun Huang. Analysis for capon and music doa estimation algorithms. In *2009 IEEE Antennas and Propagation Society International Symposium*, pages 1–4. IEEE, 2009.

[103] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. *arXiv preprint arXiv:1810.09502*, 2018.

[104] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[105] Huaiyu Li, Weiming Dong, Xing Mei, Chongyang Ma, Feiyue Huang, and Bao-Gang Hu. Lgm-net: Learning to generate matching networks for few-shot learning. In *International conference on machine learning*, pages 3825–3834. PMLR, 2019.

[106] Jaeho Lee, Jihoon Tack, Namhoon Lee, and Jinwoo Shin. Meta-learning sparse implicit neural representations. *Advances in Neural Information Processing Systems*, 34:11769–11780, 2021.

[107] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019.

[108] Yinbo Chen and Xiaolong Wang. Transformers as meta-learners for implicit neural representations. In *European Conference on Computer Vision*, pages 170–187. Springer, 2022.

[109] Mrinal R Bachute and Javed M Subhedar. Autonomous driving architectures: insights of machine learning and deep learning algorithms. *Machine Learning with Applications*, 6:100164, 2021.

[110] Ying Li, Lingfei Ma, Zilong Zhong, Fei Liu, Michael A Chapman, Dongpu Cao, and Jonathan Li. Deep learning for lidar point clouds in autonomous driving: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3412–3432, 2020.

[111] Danijela Marković, Alice Mizrahi, Damien Querlioz, and Julie Grollier. Physics for neuromorphic computing. *Nature Reviews Physics*, 2(9):499–510, 2020.

[112] Hongduan Tian, Bo Liu, Xiao-Tong Yuan, and Qingshan Liu. Meta-learning with network pruning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*, pages 675–700. Springer, 2020.

[113] Zechun Liu, Haoyuan Mu, Xiangyu Zhang, Zichao Guo, Xin Yang, Kwang-Ting Cheng, and Jian Sun. Metapruning: Meta learning for automatic neural network channel pruning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3296–3305, 2019.

Part II

Publications

Chapter 5

One-Shot Meta-Learning for Radar-Based Gesture Sequences Recognition

Gianfranco Mauro[1,2], Mateusz Chmurski[1,4], Muhammad Arsalan[1,3], Mariusz Zubert[4], Vadim Issakov[1,3].

1. Infineon Technologies AG, Am Campeon 1-15, 85579 Neubiberg, Germany
2. Department of Electronics and Computer Technology, University of Granada, 18071 Granada, Spain
3. Institute for CMOS Design, Technical University of Braunschweig, Braunschweig, Germany
4. Lodz University of Technology, Lodz, Poland

International Conference on Artificial Neural Networks (ICANN2021), Springer, Cham

- Received April 2021, Accepted June 2021, Published September 2021
- DOI: 10.1007/978-3-030-86340-1_40
- CORE2021 Rank: C

Abstract.

Radar-based gesture recognition constitutes an intuitive way for enhancing human-computer interaction (HCI). However, training algorithms for HCI capable of adapting to gesture recognition often require a large dataset with many task examples. In this work, we propose for the first time on radar sensed hand-poses, the use of optimization-based meta-techniques applied on a convolutional neural network (CNN) to distinguish 16 gesture sequences with only one sample per class (shot) in 2-ways, 4-ways and 5-ways experiments. We make use of a frequency-modulated continuous-wave (FMCW) 60 GHz radar to capture the sequences of four basic hand gestures, which are processed and stacked in the form of temporal projections of the radar range information (Range-Time Map - RTM). The experimental results demonstrate how the use of optimization-based meta-techniques leads to an accuracy greater than 94 % in a 5-ways 1-shot classification problem, even on sequences containing a type of basic gesture never observed in the training phase. Additionally, thanks to the generalization capabilities of the proposed approach, the required training time on new sequences is reduced by a factor of 8,000 in comparison to a typical deep CNN.
Keywords: Gesture recognition, Meta learning, Millimeter wave radar

5.1. Introduction

Gesture sensing technology represents a very direct and intuitive method of human-computer interaction (HCI). Under the needs of users and system interface architectures, hand movements can be identified and tracked through the use of a wide variety of sensors and detection algorithms [1]. Conventional methods for the classification of gestures involve the employment of camera sensors for optical images or time of flight (ToF) images for depth information. These sensors allow a complete and touchless understanding of the performed gestures, but they usually lead to privacy issues and poor performance in the presence of intense light [2, 3, 4]. In contrast, Radio-based sensing can be efficiently used to estimate movements and poses of subjects even through walls and obstructions [5]. Through Wi-Fi technology, the hand-pose estimation can be addressed with very high performance even in a cross-domain application, where the user's location, orientation, and environment can vary considerably [6]. However, Wi-Fi-based sensing systems require often to develop high output power in the RF range and a module in continuous working operation to exploit the functionalities. To overcome these challenges, the use of radar sensors for this application is becoming a widely adopted practice [7]. Among the various radar modulation techniques, FMCW is a particularly suitable approach, thanks to its capability of providing simultaneously accurate range and Doppler information of objects and people located in the field of view [8, 9, 10, 11]. Excellent

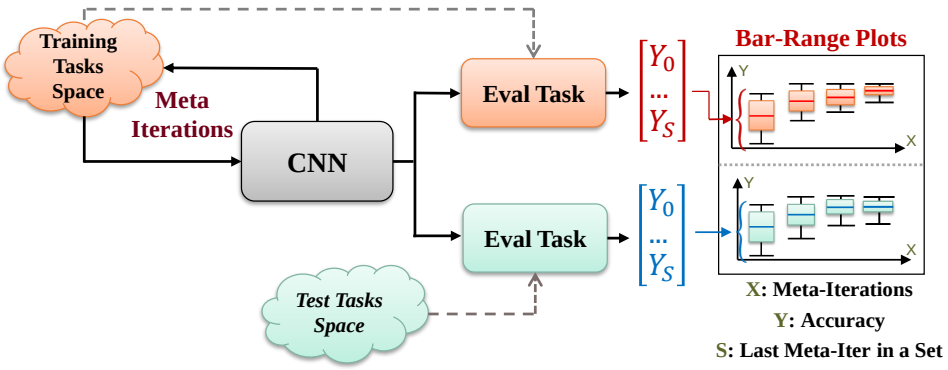


Figure 5.1: The in-training evaluation of the meta-model is performed after each meta-iteration (adaptation of the CNN to the new extracted information) on both a train and a test sampled tasks. Network generalization capability is assessed through bar plots built on batches of tasks as the meta-iterations progress.

results in the classification of gestures through range-Doppler images are achieved in [12], using the *BGT60TR13C* FMCW radar sensor [13]. The authors in [12] use the domain adaption applied to a CNN to minimize the differences among users'gestures in both learning and application stages. Through this approach, an average accuracy of 98.8 % is achieved on seven gestures performed by ten different users. Even though the state-of-the-art deep learning methods like [12] achieve excellent accuracy and robustness on radar-based gestures recognition, they demand a large amount of data to successfully train the detection algorithms [14]. This suggests that an interface based on such systems, would not be able to learn promptly how to distinguish new types of movements.

In contrast to the conventional deep learning approach, the meta-learning (Meta-L) is designed to counter the problem of huge data demand. It is based on multiple-episode few-shot optimization (tasks), which considers different learning objectives in many training steps, to extract general information from available data and efficiently solve series of problems by learning how to learn [15, 16]. The class of optimization-based Meta-L algorithms exploits the model's parameters and gradient propagation among several tasks (meta-iterations) to accomplish the generalization goal. In the inner loop of each meta-iteration, a model tries to solve an N-ways task, where N is the number of classes, that are randomly sampled from a training set of data. An example (1-shot) called *support* is then sampled for each class and used for the training. Some algorithms such as Model Agnostic Meta-Learning (MAML) [17], require additional examples per class called *query* for the evaluation of inter-tasks generalization performance after every meta-iteration.

In this paper, we suggest for the first time, the application of optimization-

based meta-learning techniques to classify sequences of hand gestures using only one sample per class. We make use of the radar range information only, in the form of RTMs of four different basic gestures, to minimize preprocessing and the CNN input data complexity. We evaluate the models with a common in-training procedure (Fig. 5.1) and test them on a sufficient number of new tasks to prove the robustness of the approach. With the use of only one sequence of gestures instance and over 50 test examples per class, we achieve an accuracy of 94 % even in the 5-ways experiments. Finally, we compare the performance results of the Meta-L approach with the ones of a conventional CNN trained on a configuration of gesture sequences. We report how the potential offline adaptation to new gesture sequences with the Meta-L model leads, in comparison with the traditional CNN, to an average training time reduction of 4 orders of magnitude.

5.2. FMCW Radar Processing

5.2.1. Radar Sensor

To capture gestures, we use the *BGT60TR13C* FMCW radar sensor [13]. The *BGT60TR13C* is equipped with one transmit (TX) and three receive (RX) channels including antennas integrated in package. During operations, the instantaneous local oscillator and reflected signals from targets are mixed and provide a resulting signal called intermediate frequency (IF) signal. As an outcome of its system power mode management and operation optimized duty-cycle, the device can run at less than 5mW for a detection range up to 5m in smart presence detection uses. Thanks to the center frequency of 60 GHz and a bandwidth of 7 GHz, this radar sensor enables a very high range resolution sensing ($\approx$ 2 cm). Moreover, time and micro-Doppler [18] analysis of the IF signal enable the discrimination of elaborate hand gestures with millimeter accuracy. The *BGT60TR13C* represents hence, a low-power and small-size solution for short-range sensing applications.

5.2.2. Time-Range Preprocessing

The data is gathered with the 60 GHz radar and then processed. It consists of RTM of four basic gestures [Down/Up, Left/Right, Rubbing, Up/-Down] with a shape of 62×32 pixels per sample. We used a single RX antenna and extracted only the range information to reduce the power consumption and to simplify the preprocessing pipeline. To obtain the representative RTMs of the gestures starting from the IF signal, we performed the following preprocessing steps. First of all, we subtracted the mean chirp value from every data frame (set of chirps). In the next step, to resolve targets over the range, we computed the first order Fast Fourier Transform (FFT) in the fast

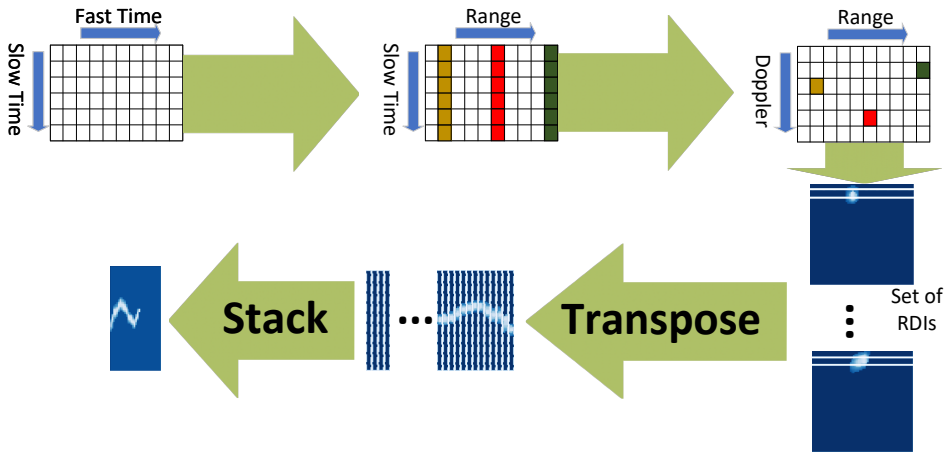


Figure 5.2: The Range Doppler images (RDI) are obtained through radar frames (IF signal) preprocessing. The lines of the RDIs with the greatest intensity are then transposed and stacked in time sequence, to obtain the RTMs.

time direction. Then, to derive the Doppler information, we performed the second-order FFT in the slow time direction.

The steps mentioned above allowed us to generate the sequence of the range-Doppler images (RDI) for every gesture. RDIs were then employed to produce the range-time images. The procedure of obtaining the range-time image is as follows:

1. identify the point with the highest intensity in the RDI;
2. cut the row in which the point with the highest intensity is localized. This row corresponds to the distance of the object from the radar in the given time step;
3. transpose each row and stack them together to form the range-time image.

The adopted preprocessing procedure in its steps is shown graphically in Fig. 5.2.

To ensure a high level of variance of the dataset, the gestures were performed by five different persons and collected in multiple environments. The experimental setup and the employed sensor (*BGT60TR13C*) are shown in Fig. 5.3.

Each gesture was recorded independently, in a time slot of 3.1 seconds. To diversify the gesture occurrence within the recording window, a random shift in time and range was also applied to every RTM. An example of RTM for each of the four basic gestures is shown in Fig. 5.4. Single gestures were

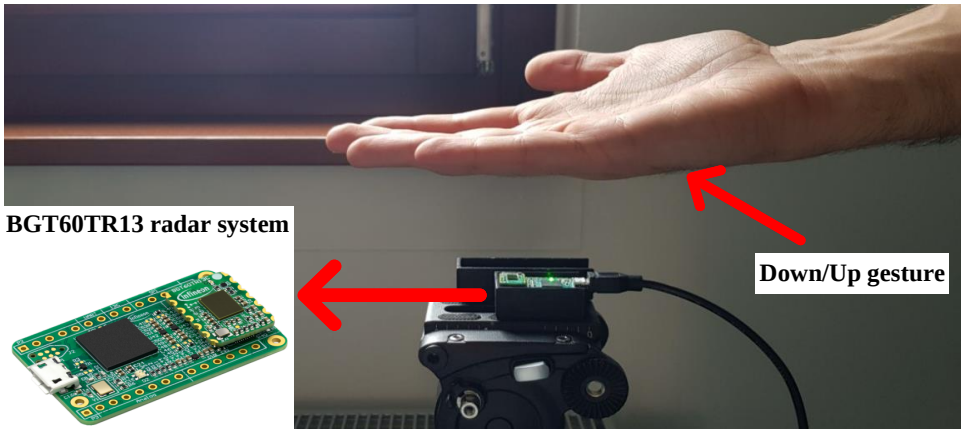


Figure 5.3: Experimental Setup (Down/Up gesture) and *BGT60TR13C*.

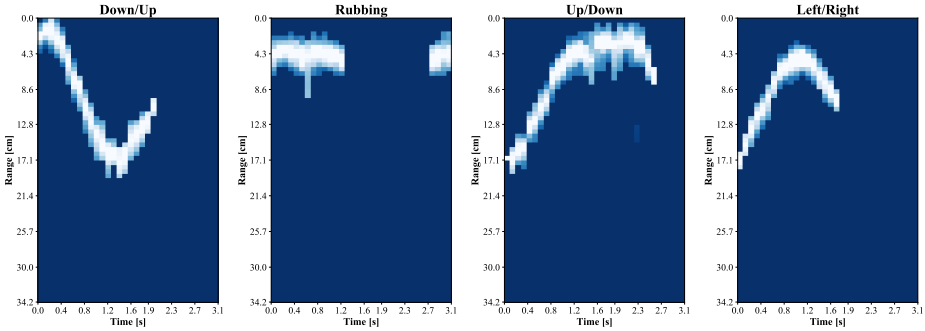


Figure 5.4: Examples of generated RTMs corresponding to the four gestures.

then stacked in channels to make sequences of two and used to generate the meta-dataset for our experiments (section 5.3.2).

5.3. Meta-Learning Based Network

5.3.1. Models and Training Procedure

As mentioned previously, we propose using an optimization-based meta-approach applied on a CNN topology, to recognize hand gesture sequences with only one sample per class in the 1-shot 2-ways, 1-shot 4-ways and 1-shot 5-ways experiments. For all the experiments, we used a CNN topology with four convolution layers of 128 filters each, for the extraction of the visual features, a kernel size 3×3 and a stride of size 2. All convolutional layers are followed by BatchNormalization, to speed up the deep network training, and by rectified linear unit (ReLU) activation function. The classification is then performed by a fully connected layer with a Softmax activation function. The chosen cost function is Sparse Categorical Crossentropy while the optimizer

is Adam. For each set of experiments, belonging to a defined number of ways, we employed three traditional optimization-based meta algorithms: Reptile [19], MAML second-order [17] and MAML first-order approximation. Additionally, we adopted a version of the second-order MAML algorithm that uses Multi-Step Loss Optimization (MSL), Derivative-Order Annealing (DA) and Cosine Annealing (CA) to stabilize inter-tasks training, as defined by the authors in [20]. The evaluation of the models is done after each meta-iteration, on a task sampled from the training set and another one sampled from a set of classes never seen by the model (test). For each S number of meta-iterations, a box-plot is built on the distribution of the obtained accuracy values. The trend of inter-tasks accuracy values in the form of box plots for sets of meta-iterations facilitates estimating the in-training learning capability of the algorithm. The employed in-training evaluation procedure is shown as part of the meta-approach schema in Fig. 5.1.

5.3.2. Meta-Dataset and Tasks Definition

Starting from the dataset D, containing the gathered data of the four basic gestures [Down/Up, Left/Right, Rubbing, Up/Down] (section 5.2), we generated a meta dataset D^{m} with 16 classes, i.e. all the possible combinations of the four initial classes. D^{m} consists of 51 samples per class, where every instance is a sequence of two RTMs that are randomly sampled from D, augmented and then stacked in the 3^{rd} dimension (channels). D^{m} is then split into two sub-datasets, $D^{\text{m-train}}$ and $D^{\text{m-test}}$. All the examples of the 7 classes that contain the basic 'Left / Right' move are included in $D^{\text{m-test}}$ so that they never appear in the training phase and therefore can be used to test the algorithm on never seen before gestures. $D^{\text{m-train}}$ contains instead all data belonging to the other 9 classes, which correspond to all the combinations of the other three basic gestures. An example of possible training and test tasks in the 1-shot 2-ways experiments, sampled respectively from $D^{\text{m-train}}$ and $D^{\text{m-test}}$, is shown in Fig. 5.5.

5.4. Experimental Results

5.4.1. Models performance

For each task in every experiment, the convolutional networks were trained for 4 epochs with inner-loop batches of size 2. The best performance results were obtained with a meta-batch of size 1 in the outer loops (inter-tasks training). An internal learning rate in the range $[5-10]e{-}4$ and an external one of $1e{-}4$ were adopted for all MAML experiments. For the MAML version that uses cosine annealing (CA), an initial outer learning rate of $2.5e{-}4$ with a decay step every 1/4 of the total meta iterations was used. For all

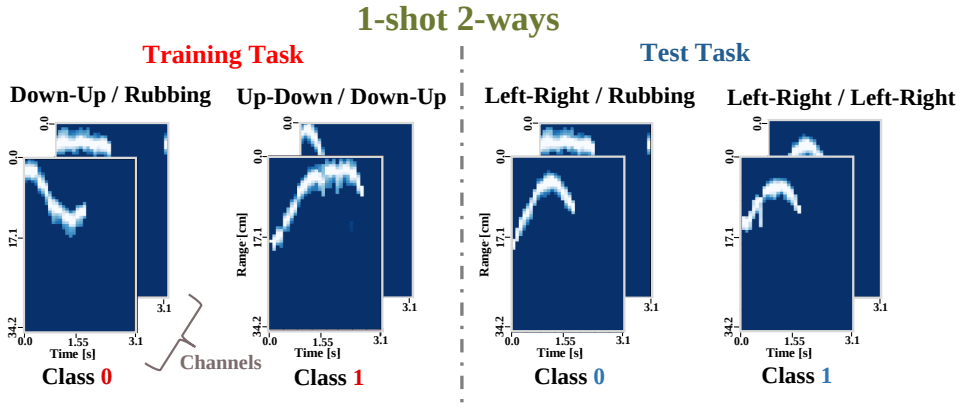


Figure 5.5: 1-shot 2-ways meta-experiments. Training and test tasks examples.

the Reptile simulations instead, an internal learning rate of $1e{-}3$ and a meta step-size for the outer loop of 0,25 have been employed. All hyperparameters, except for the outer learning rate in MAML + CA + MSL + DA, were kept constant throughout the entire meta-training procedure. The chosen number of meta-iterations was respectively 100 for the 2-way experiments, 3,000 for the 4-ways, and 10,000 for the 5-ways. Only the Reptile algorithm required 15,000 meta-iterations in the 5-way configuration to achieve a stationary inter-tasks accuracy. The inter-task generalization capacity during training was evaluated at the end of each meta-iteration following the procedure described in section 5.3.1. All experiments were performed using a Tesla P4 GPU [21, 22] and the performance of the models in terms of inter-task generalization was evaluated as the average percentage classification accuracy. All experiments were reproduced 3 times each.

Tab. 5.1 and Tab. 5.2 present respectively, the inter-task percentage median accuracy and interquartile range (IQR) values achieved with all the combinations of employed algorithms and chosen number of ways. The listed values in all the tables represent the mean values obtained over all the reproductions of each experiment for the first and last meta-tasks batches.

Fig. 5.6 shows the accuracy trend over meta-iterations as a sequence of box plots of 1,000 samples each, for the MAML + CA + MSL + DA 1-shot 5-ways experiment. The evaluation done on the training tasks is shown in red in the upper subplot of Fig. 5.6, while the evaluation on test tasks is shown in blue in the bottom subplot. The lighter colored lines in the box plots represent the median value of the accuracy (50^{th} percentile) in the set of meta-iterations, while the green triangles indicate the average value.

All the trained models were further tested on 250 tasks sampled from $D^{\text{m-test}}$. For each task, one sample per class was used to train the model and 10 for the test. This means that e.g. in the 5-ways experiments, 5 training

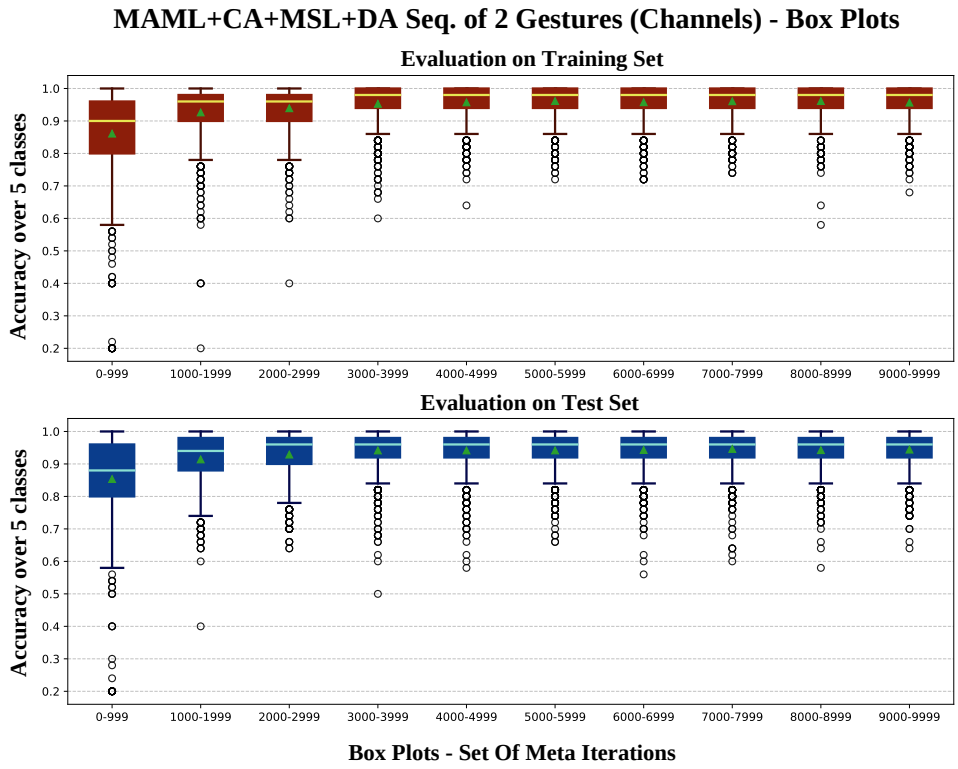


Figure 5.6: In-training evaluation of the inter-task generalization capacity for MAML + CA + MSL + DA in the 5-ways experiment. Evaluation on training tasks (upper subplot) and test tasks (bottom subplot).

samples and 50 test samples were used. The achieved percentage inter-task mean accuracy values, averaged over 3 experiments reproductions are presented in Tab. 5.3.

As can be seen numerically from the tables, the MAML + CA + MSL + DA algorithm achieves the best performances regardless of the number of ways. The application of the second gradient in MAML favors the achievement of a greater generalization and therefore of higher inter-task accuracy compared to the first-order algorithms. Furthermore, the outer-loop update, done on a query sample, increases the algorithm's robustness thus reducing the dependence on individual tasks. First-order algorithms (Reptile and MAML 1^{st} order) on the other hand, achieve very good results in the 2-way experiments but lead to significantly lower results in more complex experiments (4-ways and 5-ways). This is due to the first-order approximation of the gradient and therefore to the lack of part of the information, which becomes significant in more complex experiments.

For all the experiments, the increment in the median and mean accuracy (Tab. 5.1), and the reduction of whiskers and quartiles of box plots with

Table 5.1: Inter-task percentage median accuracy obtained on test tasks, on an average of 3 experiment reproductions for the first and last meta tasks batches. * In the Reptile 5-ways experiments (first batch: 0 - 1,499 , last batch: 13,500 - 14,999).

1-shot Experiments - Median Accuracy						
Algorithm	**2-ways**		**4-ways**		**5-ways***	
	0-24	75-99	0-299	2700-2999	0-999	9000-9999
Reptile	91.67 %	94 %	81 %	90 %	70.67 %	72.67 %
MAML 1st	94.67 %	97 %	76 %	90.67 %	72 %	85 %
MAML 2nd	95.67 %	98 %	78 %	92.67 %	86 %	96 %
MAML 2nd – CA+MSL+DA	96.33 %	98 %	82.67 %	96 %	87.33 %	96 %

Table 5.2: Inter-task interquartile range (IQR) measures, obtained on test tasks, on an average of 3 experiment reproductions for the first and last meta tasks batches. * In the Reptile 5-ways experiments (first batch: 0 - 1,499 , last batch: 13,500 - 14,999).

1-shot Experiments - Interquartile Ranges						
Algorithm	**2-ways**		**4-ways**		**5-ways***	
	0-24	75-99	0-299	2700-2999	0-999	9000-9999
Reptile	12.33 %	6.33 %	16 %	12.33 %	19.33 %	16.67 %
MAML 1st	5 %	1.67 %	20 %	13.33 %	20 %	18.33 %
MAML 2nd	4.33 %	1 %	12.33 %	9 %	16.67 %	8 %
MAML 2nd – CA+MSL+DA	3.67 %	1 %	30 %	9.33 %	16 %	6 %

Table 5.3: Inter-task percentage mean accuracy obtained on 250 test tasks for each experiment and number of ways on an average of 3 reproductions.

	1-shot Experiments - Test Accuracy		
Algorithm	**2-ways**	**4-ways**	**5-ways**
Reptile	92.59 %	86.22 %	72.36 %
MAML 1st	96.81 %	89.37 %	84.88 %
MAML 2nd	96.87 %	91.09 %	93.20 %
MAML 2nd – CA+MSL+DA	97.12 %	94.67 %	94.12 %

Table 5.4: Performance comparison of traditional and Meta-L CNNs for the 4-ways tasks. Training of both models done on a five cores CPU.

	Trad. CNN		**Meta-L CNN**	
Training samples	1000		8	4
Test samples	200		200	200
Avg. train. time	56 min	39 min	1,580 msec	400 msec
Test accuracy	98.85 %	93.67 %	98.32 %	93.47 %

progressing of meta-iterations (Tab. 5.2), represent the models ability to learn faster to solve new tasks. This means that with time, the CNN learns how to solve new tasks with better performance than before, thanks to the context information extracted from the previously faced tasks.

To exhibit the versatility of the meta-approach in adapting to new tasks, we compared our best model in the 1-shot 4-ways, with the optimized CNN defined in [23], that has been used to classify the four basic gestures dataset employing a conventional deep learning approach. In our case, we trained this traditional CNN on tasks sampled from $D^{\text{m-test}}$, using 1,000 sequences of two gestures for training and 200 for testing. Through a transfer learning approach on new tasks, this model fails to reach an appreciable accuracy value (over 85 %) despite the significant amount of training data. Consequently, each new training is done starting from a random initialization of the model parameters.

In Tab. 5.4, the average performance values of 3 independent tests of the traditional CNN are compared with the ones achieved by testing the best MAML + CA + MSL + DA model on 50 samples per class and over 250 tasks. The training of both models in this case has been done using a 5 cores CPU. The performance values achieved by the traditional CNN are presented in the relative two sub-columns of the table. The maximum achieved test accuracy and its required training time are listed in the first sub-column. The second sub-column shows instead, the time required to reach an average test accuracy comparable to that of the 1-shot Meta-L CNN. Besides, we also tested how many training shots per class are needed for the meta-model to achieve a prediction accuracy in the order of the traditional CNN.

As can be observed, the optimized CNN achieves greater accuracy on the test samples, at the expense of a large amount of data and a long adaptation time to new tasks. The meta-model, on the other hand, thanks to the pre-acquired knowledge during training, is capable of adapting to new contexts with only one sample per class and in a very short time.

5.5. Conclusion

This paper demonstrates that the use of optimization-based meta-techniques can bring significant benefits for the recognition of FMCW radar-based hand gesture sequences. The inter-tasks learning approach considerably enhances the model's ability to adapt to new potential gestures or performing users. The experimental results show how even with a single sample per sequence, it is possible to achieve an inter-task accuracy of over 94 % in the 5-way setup on new test tasks. The outcomes also highlight how the Meta-L approach can lead to an accuracy comparable to that of a traditional CNN with only a few more samples per class. Furthermore, it is shown how the adaptation of the obtained models to new tasks can take less than half of a second when performing the experiments on a 5 cores CPU. Future work will focus on the application of meta-learning for the recognition of a greater set of gestures and on an online demonstrator to test the approach.

References

[1] Mais Yasen and Shaidah Jusoh. A systematic review on hand gesture recognition techniques, challenges and applications. *PeerJ Computer Science*, 5:e218, 2019.

[2] Munir Oudah, Ali Al-Naji, and Javaan Chahl. Hand gesture recognition based on computer vision: a review of techniques. *journal of Imaging*, 6(8):73, 2020.

[3] Manju Khari, Aditya Kumar Garg, Rubén González Crespo, and Elena Verdú. Gesture recognition of rgb and rgb-d static images using convolutional neural networks. *Int. J. Interact. Multim. Artif. Intell.*, 5(7):22–27, 2019.

[4] Rytis Augustauskas and Arunas Lipnickas. Robust hand detection using arm segmentation from depth data and static palm gesture recognition. In *2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, volume 2, pages 664–667. IEEE, 2017.

[5] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7356–7365, 2018.

[6] Yue Zheng, Yi Zhang, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. Zero-effort cross-domain gesture recognition with

wi-fi. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pages 313–325, 2019.

[7] Shahzad Ahmed, Karam Dad Kallu, Sarfaraz Ahmed, and Sung Ho Cho. Hand gestures recognition using radar sensors for human-computer-interaction: A review. *Remote Sensing*, 13(3):527, 2021.

[8] Johannes Rimmelspacher, Radu Ciocoveanu, Giovanni Steffan, Matteo Bassi, and Vadim Issakov. Low power low phase noise 60 ghz multi-channel transceiver in 28 nm cmos for radar applications. In *2020 ieee radio frequency integrated circuits symposium (rfic)*, pages 19–22. IEEE, 2020.

[9] V Lammert, S Achatz, R Weigel, and V Issakov. A 122 ghz ism-band fmcw radar transceiver. In *2020 German Microwave Conference (GeMiC)*, pages 96–99. IEEE, 2020.

[10] Vadim Issakov, Andrea Bilato, Vera Kurz, Daniel Englisch, and Angelika Geiselbrechtinger. A highly integrated d-band multi-channel transceiver chip for radar applications. In *2019 IEEE BiCMOS and Compound semiconductor Integrated Circuits and Technology Symposium (BCICTS)*, pages 1–4. IEEE, 2019.

[11] Yong Wang, Aihu Ren, Mu Zhou, Wen Wang, and Xiaobo Yang. A novel detection and recognition method for continuous hand gesture using fmcw radar. *IEEE Access*, 8:167264–167275, 2020.

[12] Hyo Ryun Lee, Jihun Park, and Young-Joo Suh. Improving classification accuracy of hand gesture recognition based on 60 ghz fmcw radar with deep learning domain adaptation. *Electronics*, 9(12):2140, 2020.

[13] Saverio Trotta, Dave Weber, Reinhard W Jungmaier, Ashutosh Baheti, Jaime Lien, Dennis Noppeney, Maryam Tabesh, Christoph Rumpler, Michael Aichner, Siegfried Albel, et al. 2.3 soli: A tiny device for a new human machine interface. In *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, volume 64, pages 42–44. IEEE, 2021.

[14] Gary Marcus. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*, 2018.

[15] Joaquin Vanschoren. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548*, 2018.

[16] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169, 2021.

[17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[18] Victor C Chen. *The micro-Doppler effect in radar*. Artech house, 2019.

[19] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[20] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. *arXiv preprint arXiv:1810.09502*, 2018.

[21] Erik Lindholm, John Nickolls, Stuart Oberman, and John Montrym. Nvidia tesla: A unified graphics and computing architecture. *IEEE micro*, 28(2):39–55, 2008.

[22] Ammar Ahmad Awan, Hari Subramoni, and Dhabaleswar K Panda. An in-depth performance characterization of cpu-and gpu-based dnn training on modern architectures. In *Proceedings of the Machine Learning on HPC Environments*, pages 1–8. In Proceedings of the Machine Learning on HPC Environments. ACM., 2017.

[23] Mateusz Chmurski, Mariusz Zubert, Kay Bierzynski, and Avik Santra. Analysis of edge-optimized deep learning classifiers for radar-based gesture recognition. *IEEE Access*, 9:74406–74421, 2021.

Chapter 6

Few-Shot User-definable Radar-based Hand Gesture Recognition at the Edge

Gianfranco Mauro[1,2], Mateusz Chmurski[1,4], Lorenzo Servadei[1,5], M.P. Cuellar[3], Diego P. Morales[2].

1. Infineon Technologies AG, Am Campeon 1-15, 85579 Neubiberg, Germany
2. Department of Electronics and Computer Technology, University of Granada, 18071 Granada, Spain
3. Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain
4. Department of Microelectronics and Computer Science, Lodz University of Technology, 90924 Lodz, Poland
5. Department of Electrical and Computer Engineering, Technical University of Munich, Arcisstrasse 21, 80333 Munich, Germany

IEEE Access, Volume 10, 2022, Pages: 29741 - 29759

- Received January 2022, Accepted February 2022, Published February 2022
- DOI: 10.1109/ACCESS.2022.3155124
- Impact factor (2022): 3.9
- JCR Rank (2022): 72/158 in category Computer Science, Information Systems (Q2)

Abstract. Technological advances and scalability are leading Human-Computer Interaction (HCI) to evolve towards intuitive forms, such as through gesture recognition. Among the various interaction strategies, radar-based recognition is emerging as a touchless, privacy-secure, and versatile solution in different environmental conditions. Classical radar-based gesture HCI solutions involve deep learning but require training on large and varied datasets to achieve robust prediction. Innovative self-learning algorithms can help tackling this problem by recognizing patterns and adapt from similar contexts. Yet, such approaches are often computationally expensive and hardly integrable into hardware-constrained solutions. In this paper, we present a gesture recognition algorithm which is easily adaptable to new users and contexts. We exploit an optimization-based meta-learning approach to enable gesture recognition in learning sequences. This method targets at learning the best possible initialization of the model parameters, simplifying training on new contexts when small amounts of data are available. The reduction in computational cost is achieved by processing the radar sensed data of gestures in the form of time maps, to minimize the input data size. This approach enables the adaptation of simple convolutional neural network (CNN) to new hand poses, thus easing the integration of the model into a hardware-constrained platform. Moreover, the use of a Variational Autoencoders (VAE) to reduce the gestures'dimensionality leads to a model size decrease of an order of magnitude and to half of the required adaptation time. The proposed framework, deployed on the Intel® Neural Compute Stick 2 (NCS 2), leads to an average accuracy of around 84 % for unseen gestures when only one example per class is utilized at training time. The accuracy increases up to 92.6 % and 94.2 % when three and five samples per class are used.

Keywords: Artificial neural networks, Edge computing, FMCW, Intel Neural Compute Stick, Knowledge transfer, Meta learning, Human computer interaction, Radar, Variational autoencoder.

6.1. Introduction

HCI represents a primary field of study to enable the communication between humans and systems [1]. A classic and widely used HCI method exploits the conductivity of a user's finger or skin touch with a capacitive surface [2, 3]. Although a precise technology, this approach requires direct contact with the user and may not be versatile in specific contexts [4]. In recent years, the development of technologies such as optic or radio-frequency has radically increased the interfacing capability in all application areas [5]. Many advances in the field focus on vision-based interfacing, i.e. the use of camera sensors such as Red Green Blue (RGB) and Time of Flight (ToF) [6, 7, 8, 9]. In Fact, Camera sensors bring the advantage of touchless com-

munication. Nevertheless, Camera-based solutions lead to potential privacy issues and failures with poor light conditions in the environment. In comparison, radio-based methods are not directly affected by light and can also be used to estimate user actions through walls or barriers [10]. Wi-Fi-based systems can be robustly deployed in the HCI context even when the usage environment or the user orientation changes considerably [11, 12, 13]. Yet, Wi-Fi technology often requires the generation of high output power and a continuously running module to ensure operation. In contrast to this, radar technology, thanks to a more adaptable system power mode management, is finding increasing interest in the field of HCI applications [14]. Among the various radar modulation techniques, the Frequency Modulated Continuous Way (FMCW) is particularly suitable in the context of action recognition by providing simultaneously accurate information of the range and the velocity of targets [15, 16].

Among the various interfacing approaches, hand gesture represents a natural and easily interpretable communication mean [17, 18]. For this particular purpose, radars find wide use and can even be miniaturized and integrated into smartphones or other portable devices, such as the Google Soli [19]. State-of-the-art technology can allow hand movement sensing with high spatial resolution but must be coupled with an action recognition algorithm to enable HCI communication. Camera-based systems can find solutions based on computer vision techniques, such as skin color, skeleton, or motion recognition [20]. For radar applications, however, given the difficulty of recognizing the shape and contours of the hands, Deep Learning solutions are often adopted [21].

Machine learning finds applications in the most varied research areas, both for direct task solving and as a powerful computational tool for speeding up and modeling processes. Multiple topologies such as VGGNet [22], ResNet [23] and Inception [24] have been developed in the recent years to solve complex tasks with very high accuracy. Such networks, however, to be trained and adapted, require a fair amount of computing power and resources, which is not suitable for deployment on most edge devices [25]. Appropriate models for edge devices require specific topologies and learning processes, often leading to a trade-off between performance and adaptability. Research in the edge domain focuses mainly on two areas, namely, the topologies optimization for deployment and post-training adaptation [26]. Effective methods for reducing the size of models and the computation parameters include the use of information compression methods such as SqueezeNet [27] and depth-wise separable filters like the MobileNets [28]. Post-training model optimization can instead be achieved without important loss of performance, by employing techniques like quantization [29], factorization [30], distillation [31] and pruning [32]. Edge efficient models development has recently led to an industry movement toward such a framework. Indeed, devices with embedded

deep learning components account for a large portion of state-of-the-art HCI and Internet of Things (IoT) solutions [33]. In most of today's industrial applications of deep learning, however, models and related learning algorithms are tailor-made for specific tasks [34, 35]. While application-tuned models can achieve outstanding performance in complex and multidimensional problems, they also imply visible adaptability and interpretability weaknesses [36, 37]. The target algorithms often employ a lot of data to achieve high and robust performance. In addition, data labeling can be expensive because it may require experts, or might be sparse and depending on real-time applications [38].

A relatively new branch of machine learning, called Meta-Learning [39] has emerged to find proper solutions to problems where the adaptability on few data is essential. The idea behind Meta-Learning is to use contextual information, so-called meta-knowledge, to build a more robust model, easily adaptable to new tasks with little data. A specific subclass of meta-models called optimization-based [40] allows the transfer of meta-information between tasks via gradient method or parameters averaging. The general optimization-based approach is to learn, for a set of tasks, the best possible initialization of the parameters of a model, to make it easily adaptable in new contexts. The optimization is usually performed in the form of an episodic adaptation within two iterative steps. In base learning (inner-loop), a model learns how to solve an N-ways task, where N is the number of classes randomly sampled from the large set of training classes (if classification). In the outer loop, called meta-learning, an algorithm adapts the model following a generalization learning objective. The examples (shots) used in the inner loop are called of *support*, while the data used with the objective of generalization are called of *query*. While many meta-learning techniques rely on complex topologies and forms of gradient transmission to achieve high-performance [41, 42, 43], optimization-based techniques, given their generality, can enable the deployment of optimized models on current edge technologies.

In this paper, we propose a meta-learning optimization-based approach that enables the fast model adaptation on new gestures also at the edge. Radar-based gesture recognition in short-range applications (in the range of a few cm) represents a potential method of communication or interfacing with portable systems such as smartphones. Depending on the desired application, fast adaptation to new gestures or data may be essential. This approach can be useful not only for recognizing new action types, but also for adapting to individuals with motor disabilities or visual impairment, who are unable to perform an action in a conventional way.

We first design a radar-based setup and preprocessing suitable for the meta-learning context. Using the sensor *BGT60TR13C* FMCW [44] we gather data for a total of twenty hand gestures, performed by five users in three different environments. The collected raw data follow a definite frequency-based

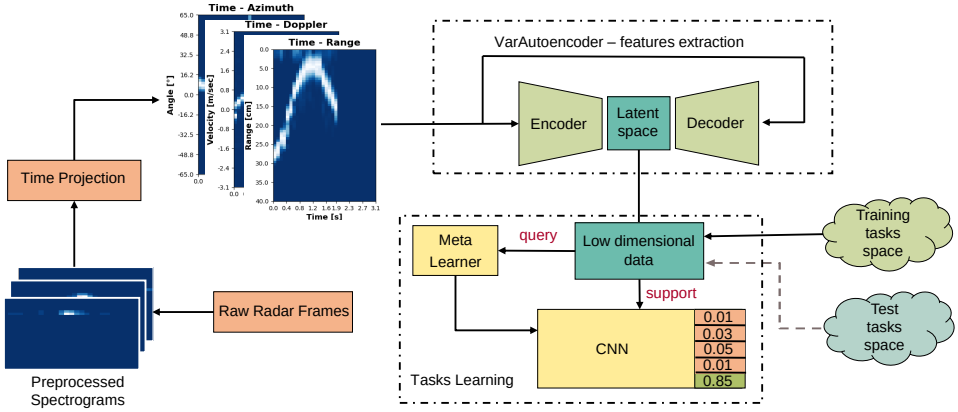


Figure 6.1: Block Diagram of the proposed model. For each gesture, the sequence of raw radar frames is initially processed in frequency. It is then elaborated and concatenated in the time domain to obtain the range, velocity, and azimuth angle of arrival information of the targets. A VAE, pre-trained on 12 training gesture classes, compresses the three-channel image into a constrained multivariate latent distribution of dimension 15. The meta-algorithm training is done on a sequence of randomly sampled tasks, exploiting the support and query data in an N-ways K-shots approach. As the meta-iterations progress, the adaptability performance is assessed on tasks sampled from the 8 test classes.

preprocessing and are then elaborated on the time axis for dimensionality reduction without relevant information loss. Then, employing Model Agnostic Meta-Learning (MAML) [45] as the base algorithm, we introduce some methods to increase the generalization capabilities of the model over new tasks. Respectively we introduce dynamic metaclass weighting (DMCW), task-specific gradient clipping (TSGC), and evaluation-based Gaussian noise summation (EGNS). We then describe how, by using part of a pre-trained Convolutional Variational Autoencoder (Conv-VAE) in the classifier, we can greatly reduce the size of the meta-model without a major loss in generalization performance. The block diagram of the proposed approach is depicted in Fig. 6.1.

We then compare the achieved results with other state-of-the-art meta-learning algorithms, showing how our solution leads to an optimal trade-off between network size, accuracy, and latency time. Finally, we perform an offline adaptation of the base model on Raspberry® Pi4 with Intel® Neural Compute Stick 2 (NCS 2), to enable the embedded application and the fine-tuning on eight defined test gestures. In this context, the training time required to tune the model to a new task on Raspberry® Pi4 and the inference time per single prediction on NCS 2 are provided. The main contributions of this paper are as follows:

1. Implementation of a proof-of-concept user-definable radar-based hand gesture recognition system at the edge. To the best of our knowledge, the first implementation at the edge in the field of radar-based user-definable gesture recognition.

2. Use of a specific preprocessing aiming at simplifying both time domain dependency and computational complexity.

3. Conceptualization of some techniques aimed at increasing the generalization capability of the algorithm on unseen gestures.

4. Design of a dimensionality reduction method, through a Conv-VAE, suitable for the optimization-based meta-learning at the edge.

6.2. Related Works

In this section, we first analyze both general and radar-based methods for hand gesture recognition. We then focus on the specific works that involve the use of little training data, such as meta-learning.

A large part of the literature focuses on the use of vision-based techniques for gesture recognition [46]. Sagayam et al. [47] proposed a method for interpreting and classifying RGB Camera-based hand gestures using a 1-D hidden Markov Model (1-D HMM). Instead of complex dynamic programming methods, a heuristic method called Artificial Bee Colony (ABC) is used for the 1-D HMM optimization. The presented algorithm leads to accurate and fast models compared to other state-based methods. The state-based approach, however, can be too slow and unsuitable for adaptation in new contexts. De Smedt et al. [48] presented a method for classifying dynamic hand skeletal data using the linear Support Vector Machine (SVM). Kinematic descriptors of gestures are extracted from the input data and then statistically and temporally coded. The pre-segmented data are then fed to the SVM for recognition. The method leads to a very low computational latency in all experiments and great performance on various datasets, but it is highly dependent on the time encoding. Liao et al. [49] illustrated a system for hand gesture-based alphabet recognition using both RGB and depth information. The Hough transform applied to the depth information is used to remove the background from the color images. The feature extraction is done through a Double-Channel Convolution Neural Network (DC-CNN). The method achieves robust performance on a large dataset but, the multi-channel approach makes it unsuitable for recognition based on other classes of sensors. Tran et al. [50] proposed a method that uses an RGB-D camera and a 3D Convolution Neural Network (3DCNN) ensemble to accurately and robustly recognize both gestures and fingertip position in real-time. Recognition is achieved through the hand skeleton-joint extracted by the

recordingsín-depth information. The model leads to a satisfactory accuracy of 97.12 % on the test data. Despite the accuracy, the method is computationally expensive and complex to adapt to new gestures, such as those featuring finger-tip oscillations. Azad et al. [51] presented a method for classifying sequences of hand depth maps by analyzing and sampling temporal information at various levels. Gesture features in the form of spatiotemporal information are derived using Weighted Depth Motion Maps (WDMM). The extracted information is further reduced by Principal Component Analysis (PCA) and classified by a single hidden layer feed-forward neural network (SLFN) with an Extreme Learning Machine (ELM). Their proposed method achieves satisfactory results in three different datasets, outperforming the results obtained by deep learning methods. Although this algorithm is less computationally complex than most deep models, its architecture is also closely related to the nature of the data and difficult to generalize to other types of input.

Other classes of sensors used for touchless gesture recognition solutions involve ultrasonic sensors and Wi-Fi technology. Das et al. [52] explored the use of ultrasonic sensors for gesture recognition as low power and low-cost alternative to optical sensors. The classification is achieved by combining a CNN and a Long Short-Term Memory (LSTM) for both spatial and temporal feature extraction. Ultrasonic sensors can represent an alternative approach to radars but, if compared to the latter, can be subject to interference phenomena and not always application-adaptable. Zheng et al. [53] presented a system for gesture recognition via Wi-Fi that enables adaptability in various domains (i.e. orientation of people, locations, and environments). The method exhibits zero-effort cross-domain adaptability employing a domain-independent body-coordinate velocity profile (BVP) estimation method. A Deep Neural Network (DNN) trained on a set of BVPs thus allows for robust recognition of as many as 15 hand gestures across domains without re-training needs. Despite the versatility of the approach, the method still requires 5,000 samples for training and is not easily adaptable to new types of gestures.

The literature on recognition using radar sensors mainly focuses on Doppler or FMCW modulated radars.

Skaria et al. [54] illustrated a method for classifying 14 types of gestures captured by a Doppler radar via deep CNN. The radar device employed is a miniaturized, low-cost dual-channel receiver model. To successfully differentiate among Doppler radar sensed gestures, the phase difference between the two antennas is exploited to infer the angle of arrival (AoA). The method shows a classification accuracy of 95 % on the test and a clear differentiation between classes. However, Doppler radars, due to their limitation in spatial resolution, find limited use for gesture recognition commonly employed for HCI. Lee et al. [55] presented a method to improve the prediction

accuracy in hand gesture recognition by *BGT60TR13C* FMCW using deep learning. The algorithm uses domain adaptation to address the problem of gesture misrecognition due to performance differences as users vary. The information extracted from the FMCW radar is frequency processed to obtain Range-Doppler Maps (RDMs). A 3D-CNN with an Inception structure processes the spatio-temporal sequence of the RDMs for classification. In parallel, an adversarial domain discriminator is used to minimize the differences between gestures performed by different users. With this method, the accuracy of 98.8 % is achieved on seven gestures performed by ten users. The domain adaptation represents a powerful generalization tool in the presence of few data but requires a related source domain rich in labels to succeed. Chmurski et al. [56] depicted how a neural network with depthwise separable convolutions can lead to high accuracy values for FMCW radar-based gesture recognition while operating in a low-power and resource-constrained environment. The model, built on eight hand gestures is optimized and deployed on the Coral Edge TPU Board. This approach although efficient is hardly adaptable to new actions.

In recent years, HCI research is evolving towards the adaptability of systems in new contexts and with little data. Rahimian et al. [57] presented a class of few-shot learning architectures for gesture recognition via electromyography. The designed approach succeeds in the generalization with only a few examples per gesture by combining temporal convolution with an attention mechanism using a meta-learning approach. The contextual information acquired with experience allows the model to adapt quickly even to new gestures which have never been observed in the training phase. Lu et al. [58] illustrated a one-shot method for gesture recognition using 3D-CNN, by exploiting transfer learning methodology from models trained with big datasets to strengthen the one-shot predictor. This approach, tested on several Vision benchmark datasets, leads to good classification and latency results. Madapana et al. [59] explored Hard Zero-Shot Learning (HZSL) on vision-based datasets for dynamic gesture recognition. The work tries to solve the classification problem by exploiting only limited training information in the form of semantic description. Although the achieved performance is far from direct data classification, this paper shows that even minimal information can lead a model to learn how to generalize.

Some work focused directly on the use of self-learning techniques for radar-based gestures. Fan et al. [60] have shown how a meta-learning approach can bring high generalization benefits for radar-based gesture recognition using FMCW modulation. The information obtained by radar for a set of seven gestures is preprocessed in the form of time maps to extract the information of range, velocity, and angle of arrival of the hands. The data is then fed in the form of tasks to an LGM-Net-based architecture [61]. The method leads to an accuracy of 97.3 % on the 2-ways task employing

5 test samples per class. However, the multi-branch structure and the elaborate learning process make it computationally complex. Zent et al. [62] have recently presented a work that focuses on gesture recognition using a Doppler sensor. The information is processed as micro-Doppler spectrograms to map over time the change in frequency caused by the hand displacement atop the sensor. Rather than learning a direct mapping between gestures and labels, the presented method, called Weighting Network, based on Relation Networks [41], learns to compare the test spectrograms with those used for training. The presented solution has the great benefits of not requiring adaptation training for new gesture types and a relatively small number of parameters. However, the architecture needs inherently to learn the direct relationship between the *support* and *query* examples in the comparison module. This characteristic, intrinsic to Relation Net-based models, can lead as exposed in [63] to lack of adaptation in the testing phase compared to other methods. Further, in [64], it has been shown how an optimization-based method can be effectively employed for HCI via FMCW radar by exploiting simplified interfacing based on hand gesture sequences and a classical CNN for classification.

6.3. System Description and Radar Preprocessing

In this section, we present the various components of the system (i.e., hardware details, operating parameters, and recording setup) and the proposed preprocessing of the data collected via radar.

6.3.1. General Overview of the Proposed Framework

The proposed framework is shown in Fig. 6.2. First of all, the raw radar signals are preprocessed to extract both frequency and time information. The data obtained for each gesture in the shape of range, Doppler, and AoA temporal maps, are then used as meta-dataset for the optimization-based meta-learning approach. Twelve types of gestures are used to train the classifier, whereas the other eight are utilized for testing. After the training process, the model is deployed through the Raspberry® Pi4 on the NCS 2 and, adapted on new test gestures to exhibit the proof-of-concept for adaptability.

6.3.2. Radar Board

In this work, gesture sensing is performed by the *BGT60TR13C* FMCW radar sensor [44], manufactured by Infineon Technologies AG. The sensor is equipped with a Transmit (TX) and tree Receive (RX) channels with an included antenna integrated into the package. The information is processed channel-wise in several steps, through the board to which the sensor is

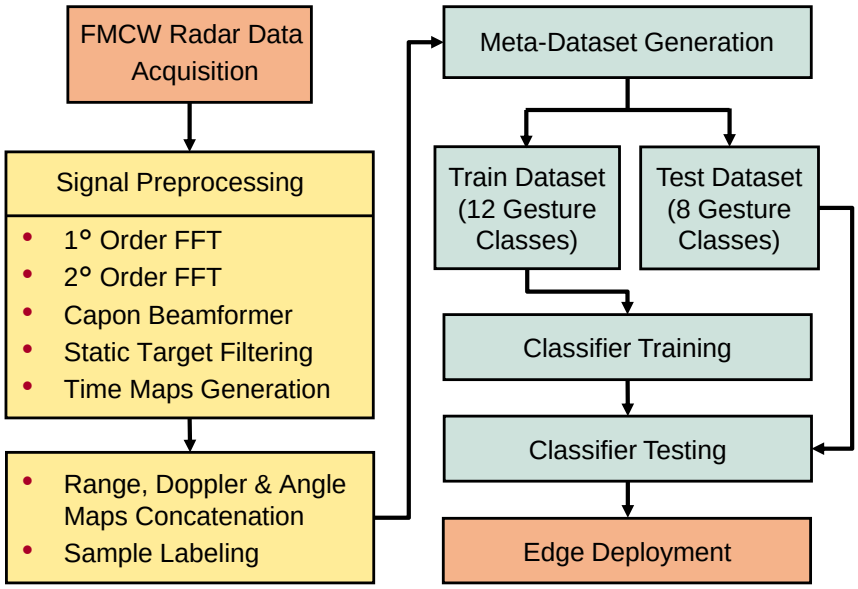


Figure 6.2: Data acquisition through FMCW radar, signal preprocessing, meta-dataset generation, and training and testing process for the proposed meta-learning-based hand gesture classifier. The orange-colored parts are hardware related. In yellow is the data processing, while in green is the classifier part. The frequency analysis is enabled by Fast Fourier Transform (FFT).

connected Fig. 6.3. The operating principle of the sensor relies on linear frequency modulation of continuous waves. The TX transmits periodic signals called chirps and, the RXs receive signals reflected from the targets located in front of the sensor. During operations, the instant local oscillations are mixed with the reflected signals and result in an output signal called the Intermediate Frequency (IF). The IF signal is then passed to a baseband chain and digitalized through an analog-to-digital converter (ADC) with 12-bit resolution.

The *BGT60TR13C* is a miniaturized solution with a center frequency f_0 of 60 GHz and a bandwidth of about 6 GHz that enables a high range solution ($\approx$ 2 cm). The phase analysis of the IF signal, exploiting the micro-Doppler effect [65], can also enable the discrimination of displacements with millimeter accuracy. Thanks to the 3 RX channels orthogonal to each other, the radar enables the estimation of both azimuth (between 65 and -65 degrees) and elevation (between 45 and -45 degrees) AoA of targets. This system also features power mode management and an operation-optimized duty cycle to reduce power consumption to only 5 mW for applications within the 5 m range. The *BGT60TR13C* represents so, a low-power and miniaturized solution for short-range sensing applications.

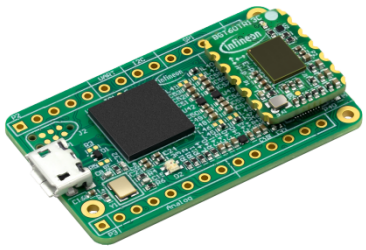


Figure 6.3: *BGT60TR13* Radar System. The radar sensor, is mounted on top of the board.

6.3.3. Radar Parameters Configuration

The *BGT60TR13* system allows to transmit for each so-called radar frame, a sequence of N_c chirps with a single signal duration time t_c along the slow-time dimension. Each chirp also consists of a number n_s of samples along the fast-time dimension. The transmitted signals use the saw-tooth wave function modulation to enable a linear behavior during the chirp rise phase. For an FMCW radar, the range resolution Δr and the maximum detection range R_{max} can be derived through the following formulas:

$$\Delta r = \frac{c}{2B_w} \tag{6.1}$$

$$R_{max} = \frac{\Delta r}{2} \cdot n_s \tag{6.2}$$

where c is the speed of light and B_w represents the frequency bandwidth around the central f_0 frequency. A bandwidth of 6 GHz, between 57 GHz and 63 GHz, has been chosen to enable a high range resolution of about 2.5 cm. The number of samples per chirp has been set to 32 for enabling the detection of targets up to a range of 40 cm. Further, an ADC sampling frequency F_s of 2 MHz has been chosen not to limit R_{max} because of signal conversion. The velocity resolution Δv and the maximum detectable velocity in a given direction V_{max} can be computed as:

$$V_{max} = \frac{c}{4f_0 t_c} \tag{6.3}$$

$$\Delta v = \frac{V_{max} \cdot 2}{N_c}. \tag{6.4}$$

A number of 64 chirps per frame N_c with single signal duration time t_c of 390.4 μs, has been chosen to allow a V_{max} of about 3.14 m/s and a Δv of about 9.8 cm/s respectively. The parameters used for radar configuration in the hand gesture sensing application are in Table 6.1.

Table 6.1: Radar Sensor Parameters Configuration.

Symbol	Quantity	Value
f_0	center frequency	60 GHz
B_w	bandwidth	[57 – 63] → 6 GHz
F_s	sampling frequency ADC	2 MHz
N_c	number of chirps	64
t_c	chirp time duration	390.4 µs
n_s	samples per chirp	32
$Azi.AoA$	azimuth angle of arrival	-65 – 65 deg
fps	frames per second	10

6.3.4. Radar Signal Preprocessing

The raw sensed radar data are not easily interpretable due to spatial resolution constraints and the influence of noise and environment surrounding the targets. While it may be possible to develop an application based on raw data as input, this would involve the training on a large amount of data that only partially contains the target information. In this work, we propose to process the signals first in frequency to extract and separate the shifts in range and velocity caused by the hands located in front of the sensor. For each detected gesture, the information is processed frame-wise and then concatenated in time to project the range and velocity contents in the 2-D plane. In such a way, Range Time Maps (RTM) and Doppler Time Maps (DTM) are generated. Exploiting the signal sensed by two RX channels, the AoA azimuth is also estimated via Capon beamformer algorithm [66]. The azimuth information is then processed and projected on the temporal plane for each frame to form Angle Time Maps (ATM).

6.3.4.1. Single Frame Preprocessing

For this application, the IF signal $S_{IF}(n)$ for each of the three available RX channels $n \in N_{RX}$ is employed to build a frame. For each n channel, the data are arranged in a 2-D matrix with slow time for the x-axis (raws) and fast time for the y-axis (columns). For each frame, by frequency analysis using Fast Fourier Transform (FFT), the Range-Doppler Image (RDI) is first calculated. The AoA azimuth is then estimated using the Capon algorithm to build the Range Angle Image (RAI). Fig. 6.4 depicts the various preprocessing steps used to obtain frame-wise RDI and RAI.

The first part of the preprocessing consists of the following steps.

1. First the mean values, computed over the fast time are subtracted along the slow time axis.

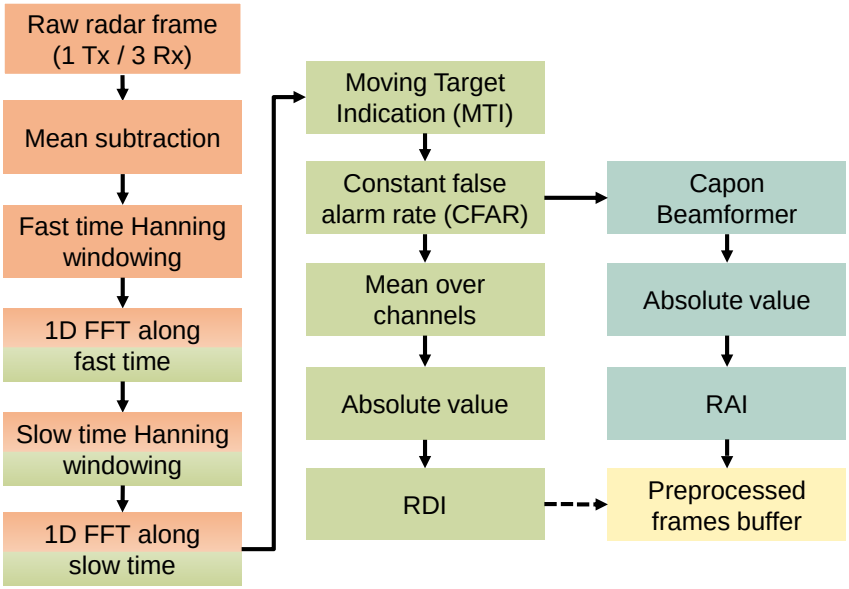


Figure 6.4: Diagram illustrating step by step the preprocessing used on each radar frame. In orange are shown the operations performed in the time domain, in green those done in the frequency domain, in blue the AoA computation.

2. The data are then multiplied with a Hanning window along the fast time to minimize spectral leakage effects for frequency analysis.

3. The 1-D FFT along fast time is executed to extract the range information.

4. Hanning windowing is applied along the slow time axis.

5. The 1-D FFT along slow time is performed on data to extract the velocity information.

6. The moving target indication (MTI) is next applied to discriminate targets against unwanted background information, aka clutter (6.5).

$$S_{IF}(n) = \alpha \cdot S_{IF}(n) + (1 - \alpha) \cdot \overline{S_{IF}}(n) \tag{6.5}$$

where α is a parameter in the range $[0 - 1]$ set to 0.9, and $\overline{S_{IF}}(n)$ the updated moving average for each frame.

7. A Constant False Alarm Rate (CFAR) algorithm is used for each channel n to filter the frequency peaks and increase the Signal-to-Noise Ratio (SNR).

To further increase the SNR for the RDI computation, the absolute value of the average of $S_{IF}(n)$ over the N_{RX}, as shown in (6.6).

$$RDI = \left| \frac{1}{N_{RX}} \cdot \sum_{n=0}^{N_{RX}} S_{IF}(n) \right| \tag{6.6}$$

After using CFAR, the $S_{IF}(n)$ associated with the two RX channels placed in the horizontal plane is processed by Capon beamforming for the AoA computation. The absolute value is then calculated and the RAI is generated.

6.3.4.2. Gesture Sensing and Time Projection

Gesture sensing begins when an average $\overline{S_{IF}}$ for the three RX channels is higher than a defined threshold, which is computed every time the sensor is turned on for a new recording session (i.e. new environment or new user). The threshold is determined as the average value of the last 20 collected frames (2 s) and it is used for comparison at every timestamp during operation. A gesture is considered gathered when the threshold is not exceeded for 5 consecutive frames. The recording window has a length of 3.1 s and therefore contains up to 32 frames for every performed action.

The stored frames, are then preprocessed in the form of RDI and RAI and mapped into a lower-dimensional space to compute the RTM, DTM and ATM. For each RDI or RAI, belonging to a sequence of matrices definable as $A : \{1, ..., m\} \times \{1, ..., m\} \times \{1, ..., t\} \to \mathbb{R}$, where $t \geq 1$, the goal is to find the index (x, y) corresponding to the maximum value $a_{x,y}^{max}$.

$$A_{m,n} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

where $m \times n$ represents the range and Doppler dimensions for the RDI and range and angle dimensions for the RAI. The information, corresponding to the distance and velocity of the target from the sensor, is extracted by taking from the RDI the $Col_x(A)$ and the $Row_y(A)$ respectively. The AoA azimuth is instead extracted from the RAI by taking the $Row_y(A)$. The concatenation of the obtained rows or columns for the whole gesture duration leads to the generation of the RTM, DTM, and ATM. Fig. 6.5 illustrates graphically the principle of range information extraction given a sequence of frames. Each hand pose is represented by 3-channel information (RTM, DTM, and ATM). The gestures collected with fewer than 32 frames are expanded via zero padding at the end of the time sequences. All instances are normalized channel-wise in the [0 – 1] range.

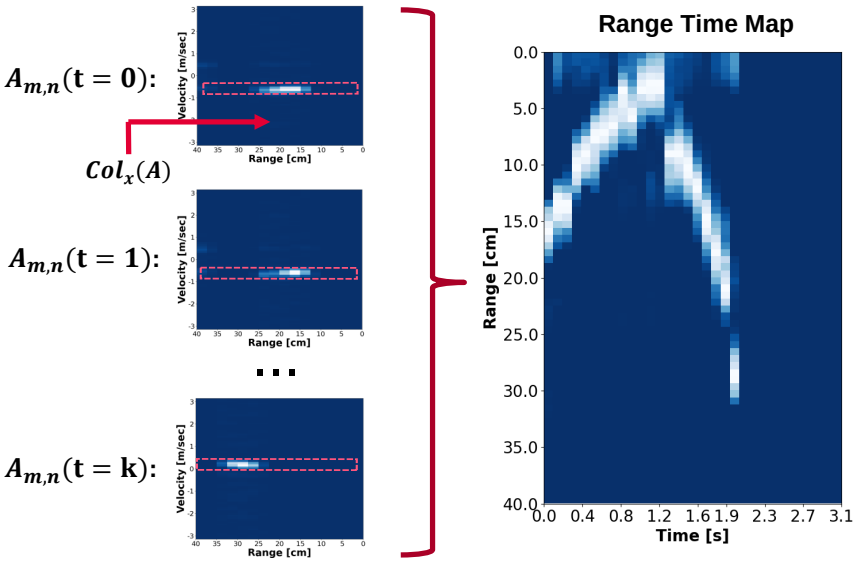


Figure 6.5: Example of time projection for an RTM generation.

6.3.5. Recording Setup

In this work, we sensed gestures via radar for a total of twenty classes. The recording setup for data collection consists of a Raspberry® Pi4 and the *BGT60TR13* board. The radar board is mounted on a tripod through a 3D-printed case. With the defined radar configuration, a maximum detection range of 40 cm implies the potential use as short-range application only. Such setup is therefore meant for handheld or turnstile gesture recognition interfaces. The setup in its components is depicted in Fig. 6.6. The actions have been performed by a total of five users and in three different environments (office, hall, and outdoor). These specific environments have been chosen among several possible, as they represent three contrasting application contexts. In the office, the presence of static furniture and devices placed in the radar's field of view can result in added reflections and subsequent noise in the preprocessed signals. The hall and outdoors instead, represent two wide environments where, in first approximation, only the arm and potentially the body of the subject performing the gesture fall within the field of view of the radar. The data were collected partly outdoors to avoid possible dependencies from secondary reflections given by devices and metal ducts placed in the hall environment. The consent has been obtained from users prior to data collection and as much anonymity and privacy as possible were maintained during the data collection and processing phases. Individuals of varying height [1.60 – 1.85] m and age [25 – 40] years with no relevant motor or visual impairments have been engaged in the experiment. The only information given to the users before performing the gestures were the radar

orientation, and the maximum duration of the gestures of 3.1 s. The data have not been saved in online archives and/or published. The chosen gestures are those most commonly employed for HCI in touchless applications. Only in the final test phase, to demonstrate the offline proof-of-concept of

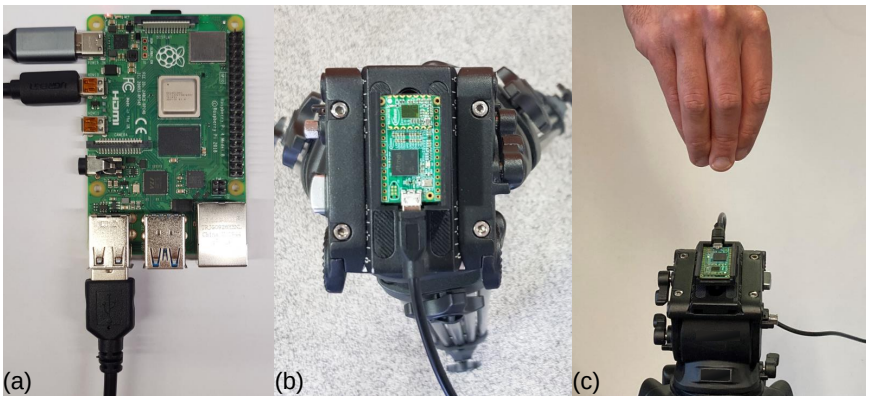


Figure 6.6: Recording setup for gestures sensing. (a) shows the Raspberry® Pi4 employed for data recording. (b) depicts the BGT60TR13 radar board on the tripod. (c) shows an example of performed action for the class rubbing".

the system's adaptability to new gestures, the developed model is deployed on Raspberry® Pi4 and NCS 2. This setup is shown in Fig. 6.7.

Figure 6.7: Recording setup for the offline proof-of-concept of the system generalization capability at the edge. The Raspberry® Pi4 is used for data preprocessing, model adaptation and script running. The NCS 2 enables the deployment of the developed meta-learning model for a specific setup.

6.3.6. Gestures Dataset

For the meta-learning approach, the gestures are split by classes between a meta-training $\mathcal{D}^{m-train}$ and meta-test $\mathcal{D}^{m-test}$ sets. Fig. 6.8 illustrates the

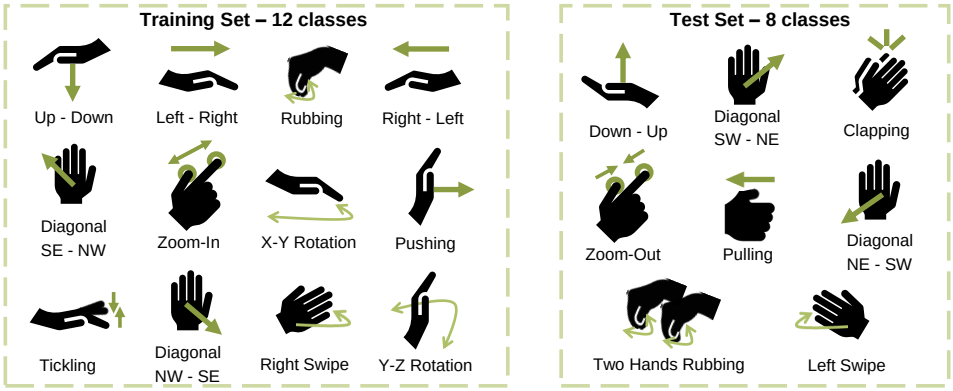


Figure 6.8: Gestures vocabulary for the meta-training and meta-test datasets. N, S, W and E represent the cardinal points.

twelve training and eight test gestures, respectively. The division of gestures has been performed randomly, with the only constraint to keep, in the two datasets, the sets of gestures that are opposite to each other. A t-distributed Stochastic Neighbor Embedding (t-SNE) representation of the gestures in two components is shown in Fig. 6.9.

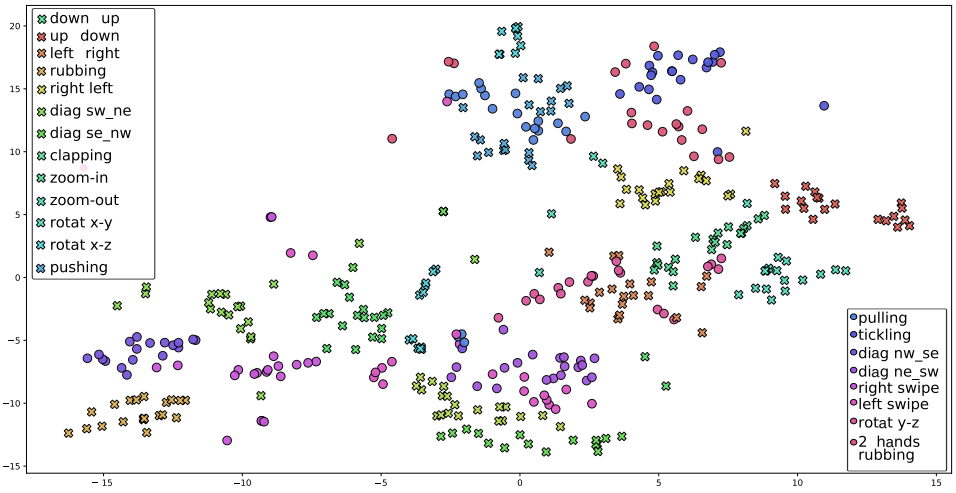


Figure 6.9: 2-D components t-SNE representation of the twenty gestures of the dataset. Classes belonging to $D^{m-train}$ are represented with a cross marker. Classes belonging to D^{m-test} are represented by a point marker.

Extracting both range and azimuth information is crucial for correctly distinguishing some gestures from others. Examples where RTM and ATM clearly allow a distinction between two classes are shown in Fig. 6.10 and Fig. 6.11, respectively. Velocity information can improve the separation between classes, especially concerning the spatial plane in which the gestures

are performed. In addition, such information can help distinguish actions characterized by local finger oscillations, such as rubbing and tickling Fig. 6.12.

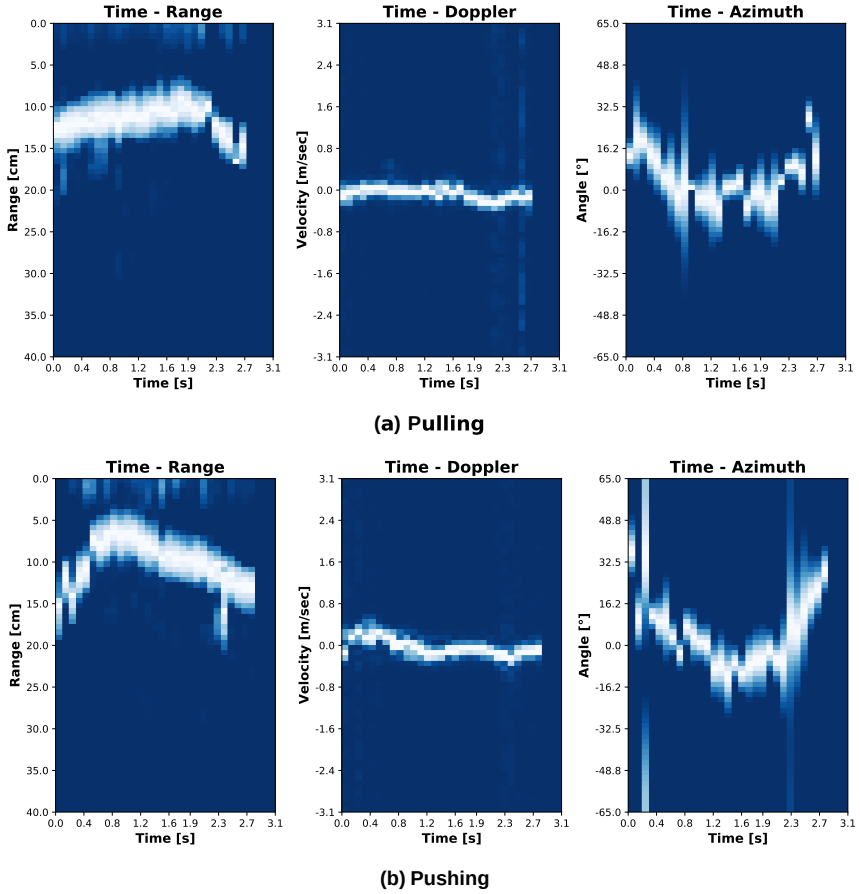


Figure 6.10: Comparison of RTM, DTM and, ATM between (a) Pulling, and (b) Pushing. In this example, the range information allows a clear distinction between the two classes.

6.4. Proposed Method

In this section, we propose our approach, which belongs to the class of optimization-based meta-learning algorithms. We first introduce some methods to increase the model's generalization capability in comparison to the state-of-the-art. We then present the adopted CNN topology and the benefits of using a pre-trained Conv-VAE as a backbone in the meta-learning phase to reduce the number of parameters.

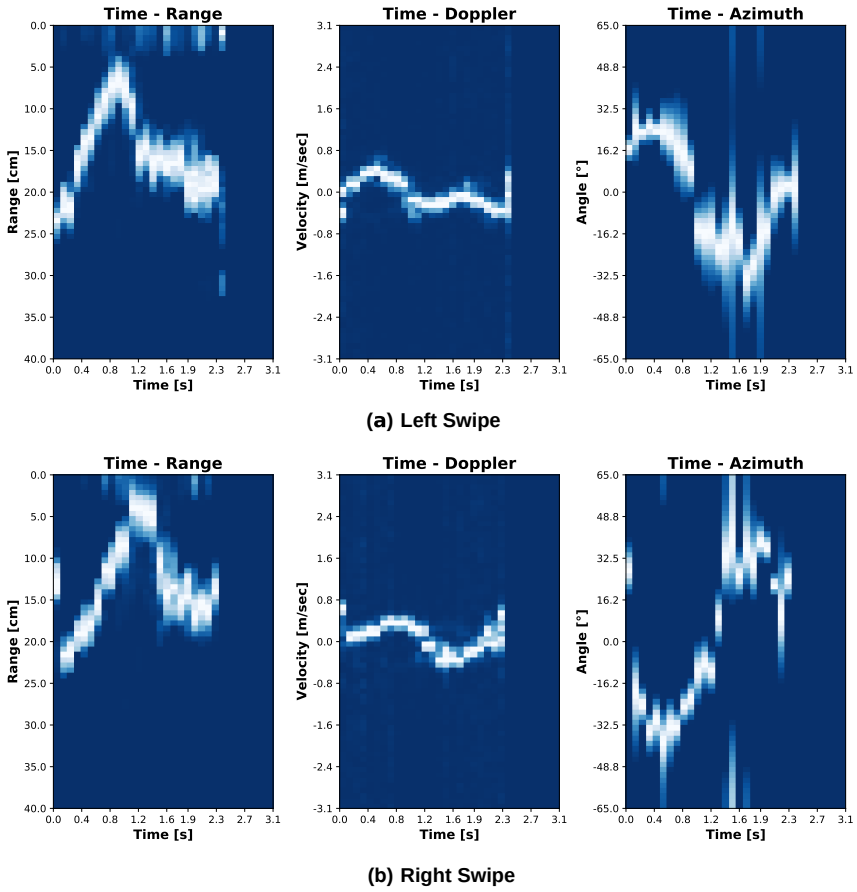


Figure 6.11: Comparison of RTM, DTM and, ATM between (a) left swipe, and (b) right swipe. In this example, the azimuth information allows a clear distinction between the two classes.

6.4.1. Optimization-based Meta-Learning

In a conventional optimization-based meta-learning approach for deep learning, the optimization consists of two iterative steps performed over the distribution of tasks $p(\mathcal{T})$, to train a model represented by a parametric function f_θ with parameters θ. The two optimization steps are the following:

1. In *base-learning*, for a batch of N tasks, an inner learning model $f_{\theta'_n}$ with parameters θ'_n, tries to solve each task $\mathcal{T}_n$, given a dataset $\mathcal{D}_{\mathcal{T}n}$ and a task related loss function to minimize $\mathcal{L}_{\mathcal{T}n}(f_\theta)$.

2. In *meta-learning*, an outer algorithm makes use of the information obtained through back-propagation of the gradient in the inner learning phase to update the internal algorithm. The model trained during base learning also minimizes an outer loss function $\mathcal{L}_{ext}(f_{\theta'_n})$.

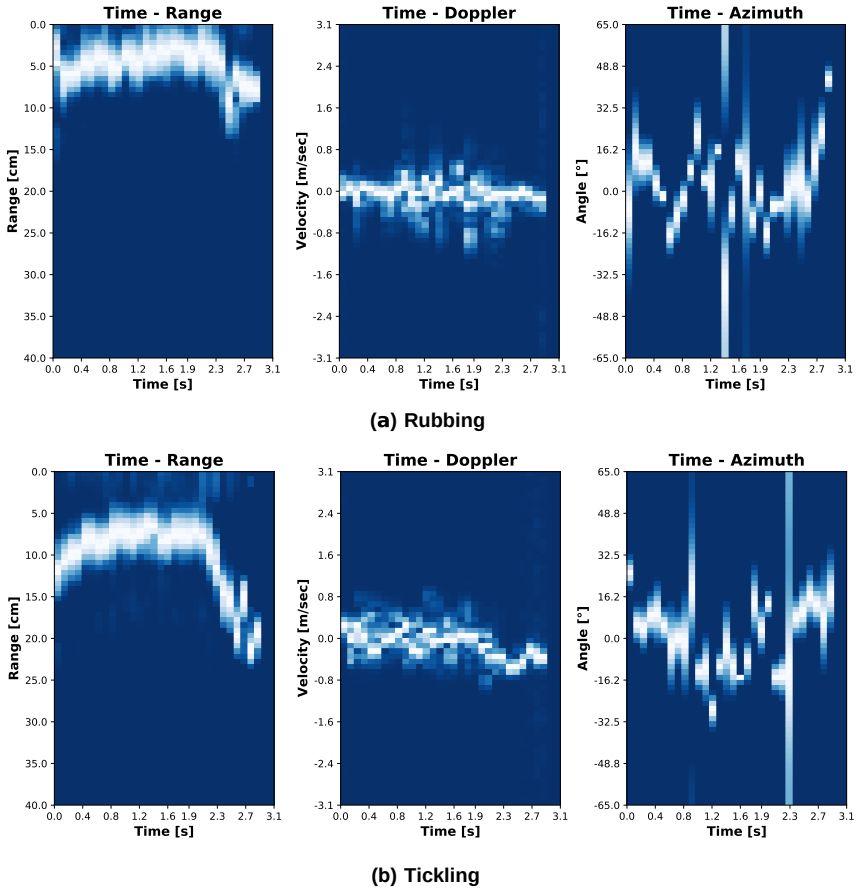


Figure 6.12: Comparison of RTM, DTM and, ATM between (a) rubbing and (b) tickling. Local oscillation caused by finger movement in the velocity profile can be noted for both classes.

If the loss function defined for the task is differentiable, the internal optimization is often performed by Stochastic Gradient Descent (SGD) in K batches of training examples belonging to $\mathcal{D}_{\mathcal{T}n}$. The θ' parameters are computed as:

$$\theta'_n = \theta - \gamma \cdot \sum_{k=1}^{K} \nabla_\theta \mathcal{L}^{(k)}_{\mathcal{T}n}(f_\theta) \tag{6.7}$$

where γ is the inner loop learning rate of the meta-algorithm. In [45], Finn et al. present a very general method called Model Agnostic Meta-Learning (MAML) where the meta-optimization across tasks is also performed via SGD, by minimizing the function $f_{\theta'_n}$ with respect to θ, for each single task

or N tasks sampled from $p(\mathcal{T})$.

$$\begin{aligned} &\min_{\theta} \frac{1}{N} \cdot \sum_{n=1}^{N} \mathcal{L}_{ext}^{(n)}(f_{\theta'_n}) \\ &= \frac{1}{N} \cdot \sum_{n=1}^{N} \mathcal{L}_{ext}^{(n)}(f_{\theta - \gamma \cdot \sum_{k=1}^{K} \nabla_{\theta} \mathcal{L}_{\mathcal{T}n}^{(k)}(f_{\theta})}) \end{aligned} \tag{6.8}$$

$$\theta \leftarrow \theta - \beta \cdot \frac{1}{N} \cdot \nabla_{\theta} \sum_{n=1}^{N} \mathcal{L}_{ext}^{(n)}(f_{\theta'_n}) \tag{6.9}$$

where β in (9) is the outer loop learning rate. In MAML for few-shot supervised learning, two different data sets are defined for each task $\mathcal{T}_n$. Support samples $\mathcal{D}_n$ for base learning and query $\mathcal{D}'_n$ for the inter-tasks generalization step in the meta-learning phase. As can be seen in (8), meta-gradient involves a gradient through a gradient and can lead to instability during training as well as resulting computationally expensive. Antoniou et al. [67] present various modifications to the MAML to enhance the learning stability and also the generalization capability.

In our work, we adopt MAML as the base algorithm, with a task batch size N of 1 and, we exploit some of the methods presented in [67] to improve the training stability. Specifically, we leverage the following contributions:

- **Multi-Step Loss Optimization (MSL)**: instead of minimizing the outer loss function after the completion of all base learning steps for support set task $\mathcal{D}_n$, we do an update after each inner-epoch $i \in I$, composed of K batches, using $\mathcal{D}'_n$. Specifically, we exploit a set of importance weights v_i that enables a higher loss contribution for the latest i in I.

$$\theta \leftarrow \theta - \beta \cdot \nabla_{\theta} \sum_{i=1}^{I} v_i \sum_{k=1}^{K} \mathcal{L}_{ext}^{(k)}(f_{\theta'_k}) \tag{6.10}$$

  In addition, as the meta-iterations performed on the distribution of tasks $p(\mathcal{T})$ progress, the relative weights of early epochs are decreased and, those of the late epochs are increased. This strengthens the ability to learn from every individual $\mathcal{T}_n$ task without potentially destabilizing learning. In comparison to the method proposed in [67], where the update of the outer loss is performed after each step towards the support set task, we suggest an update after each inner-epoch. This leads to a trade-off between intra-task learning steps and computational complexity.

- **Derivative-Order Annealing (DA)**: the use of the second-order gradient involves some computational expenses and can make the optimizer inefficient and unstable during the early training phase of MAML.

To overcome these problems, we anneal the derivative order in the first 50 meta-iterations by exploiting the first-order gradient information only.

- **Cosine Annealing of Meta-Optimizer Learning Rate (CA)**: to fine-tune the optimization via the outer algorithm as the meta-iterations progress, we apply a cosine annealing scheduling on the optimizer. This yields an increase in generalization performance without impacting the per task computation $\mathcal{T}_n$.

We besides propose some methods that can increase the generalization capability of MAML without bringing any increase in computational complexity in evaluation and testing. Respectively, for this purpose, we present the Dynamic Meta Class Weighting (DMCW), Task-Specific Gradient Clipping (TSGC), and the Evaluation-based Gaussian Noise Summation (EGNS).

6.4.1.1. Dynamic Meta Class Weighting

In a task learning approach with only a few data, a model can easily overfit the training instances leading to weak classification performance on the testing instances. Few examples per class may not be informative enough for the description and lead to significant misclassifications in testing. One way to counter this is to use in the inner loop, for each task $\mathcal{T}_n$, a set of class weights $\forall\ c \in C$, where C represents the number of ways. Specifically, we propose to compute after each inner-epoch, for each $c \in C$, a weight v_c which is inversely proportional to the number of correct predictions. The idea is to sample for each task $\mathcal{T}_n$, a balanced set of examples $\mathcal{D}_c \neq \{\mathcal{D}_n; \mathcal{D}'_n\}$ on which each inner epoch performance can be dynamically evaluated. For a given class c, with corresponding M weighting examples x_m, the normalized weight $\overline{v}_c$ in the range [0–1] is computed as follows:

$$\overline{v}_c = \frac{1}{\sum_{c=1}^{C} v_c} \cdot \sum_{m=1}^{M} (\hat{y}_m - y_m) \tag{6.11}$$

where y_m represents each instance-associated label, $\hat{y}_m$ the predicted label after every inner-epoch and, v_c the computed weights before normalization. The resulting $\overline{v}_c$ weights are used both in the *base learning* and the *meta-learning* updates after each batch k in K. Respectively:

$$\theta'_n = \theta - \gamma \cdot \sum_{k=1}^{K} \overline{v}_c^{(k)} \cdot \nabla_\theta \mathcal{L}_{\mathcal{T}n}^{(k)}(f_\theta) \tag{6.12}$$

and for the *meta-learning* update, through MSL:

$$\theta \leftarrow \theta - \beta \cdot \nabla_\theta \sum_{i=1}^{I} v_i \sum_{k=1}^{K} \overline{v}_c^{(k)} \cdot \mathcal{L}_{ext}^{(k)}(f_{\theta'_k}) \tag{6.13}$$

Each inner update improves intra-task classification performance by bringing more attention to minimizing $\mathcal{L}_{\mathcal{T}n}$ on classes whose examples have been poorly classified. In addition, the outer update allows inter-task propagation of the information obtained with the weights $\overline{v}_c$ to improve generalization performance.

6.4.1.2. Task-Specific Gradient Clipping

Task training performed with little data for a given number of epochs I brings benefits in some cases but can also lead to gradient explosion and instability in others. The model can so overfit on a given task, making generalization to others less effective. One solution to this is performing gradient clipping for the intra-task updates when the gradient exceeds a threshold, as presented by Pascanu et al. [68]. In our case, we suggest using clipping in the intra-task phase, for each batch k in K on the when the gradient $\mathbf{g}$ computed for $\mathcal{L}_{\mathcal{T}n}$ exceeds a certain threshold h:

$$\begin{cases} \mathbf{g} = \frac{\partial \mathcal{L}_{\mathcal{T}n}(f_\theta)}{\partial \theta} \ , \\ \mathbf{g} \leftarrow \frac{\mathbf{g} \cdot h}{||\mathbf{g}||} \ , \qquad if \ ||\mathbf{g}|| > h \end{cases} \tag{6.14}$$

where $||g||$ represents the L2 norm computed on the gradients. We propose further not to use gradient clipping for the intra-task update on queries via $\mathcal{L}_{ext}$. By doing so, the query update grants a higher contribution to the whole optimization-based procedure.

6.4.1.3. Evaluation-based Gaussian Noise Summation

Training on a sequence of tasks for a large number of meta-iterations can make the algorithm too specific on $\mathcal{D}^{m-train}$ and thus decreasing the generalization capability on $\mathcal{D}^{m-test}$ leading to the so-called meta-overfitting. One way to counteract such behavior on $D^{m-train}$ is to increase the complexity of the task when the performance becomes very high. One way to make a task n more complex is to add Gaussian noise to the examples x_n in $\mathcal{D}_n$ or to their embedded representations as to the output of the hidden layers of the model. Specifically, we propose to sum to the output of various depths of the model, random Gaussian noise in the interval $[-\sigma \ ; \ \sigma]$ from the distribution $\mathcal{N}(\mu, \sigma^2)$ generated for each batch k in K. This Gaussian noise is activated for a new training task only when the validation accuracy, performed on a sequence of tasks, sampled by $\mathcal{D}^{m-train}$, exceeds a defined threshold.

6.4.2. Proposed Topologies

For the optimization-based meta-learning approach, we propose the use of two topologies. First a traditional one, consisting of sets of convolutional

layers for features extraction. Then, a structure that uses part of a Conv-VAE as a backbone to considerably reduce the number of parameters in the overall topology. For both neural networks, the goal is, given a task $\mathcal{T}_n$, to map the sequence of RTMs, DTMs, and ATMs belonging to a gesture to the respective class.

6.4.2.1. Convolutional Neural Network

The first topology consists of three convolutional layers with the final dense layer. The convolutional layers use 128, 256, and 512 filters respectively, with a kernel size 3×3 and a stride of 2. Each of these layers is followed by batch normalization, to increase the training stability for each batch k, and by the ReLU activation function. A Flatten layer and a Dense layer are attached to the last of the three convolution blocks. The Dense layer output neurons correspond to the number of classes in the experiment. The classification is enabled through the Softmax activation function, which maps the output vector into a classes probability distribution. The topology is depicted in Fig. 6.13.

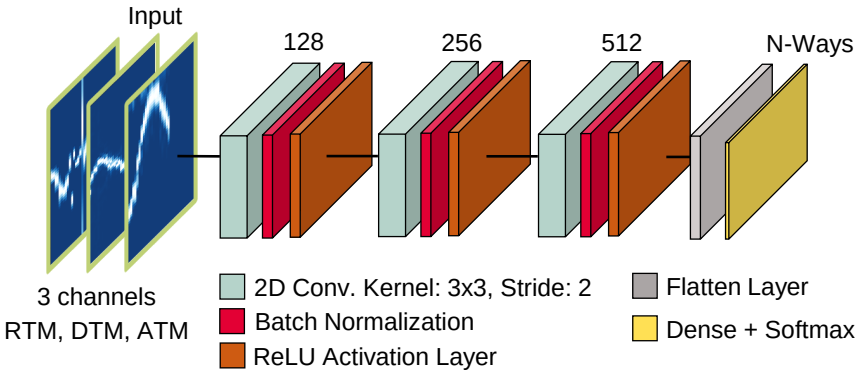


Figure 6.13: CNN topology. For each gesture, consisting of RTM, DTM, and ATM information in-depth channels, features are extracted from three blocks of convolutional layers. A final dense layer with Softmax activation enables the classification. The number of filters per convolution is noted above the respective blocks.

6.4.2.2. Conv-VAE and Dense

The second topology exploits part of a Conv-VAE, pre-trained on $\mathcal{D}^{m-train}$, to significantly squeeze the input size. The Conv-VAE compresses the three-channel information (RTM, DTM, and ATM) into a constrained multivariate latent distribution of dimension 15. The Encoder part of the Conv-VAE model is then extracted and concatenated to a sequence of Dense layers for task

training. The Dense layers consist of 256, 128, and N output neurons respectively, corresponding to the number of ways for the experiment. Also for this topology, the outputs of the last layer are mapped in a classes probability distribution through Softmax. The layers extracted from the Conv-VAE are also trained during optimization-based meta-learning for the N-ways classification objective. The topology is shown in Fig. 6.14.

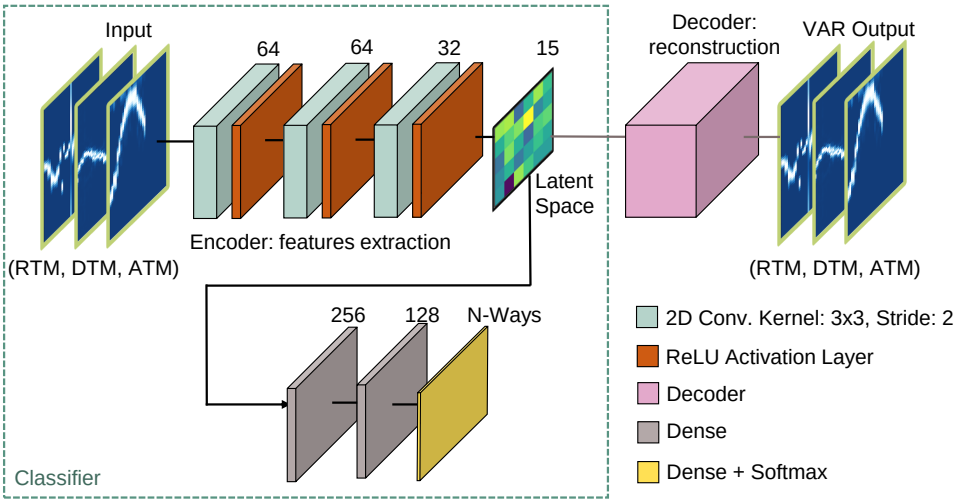


Figure 6.14: Conv-VAE and Dense topology. To significantly reduce the number of parameters compared to the convolutional model, the classification is done by exploiting the encoder of a Conv-VAE pre-trained on $\mathcal{D}^{m-train}$. For the categorical classification, three Dense layers connected to the final layer of the encoder (latent space) are used. The number of filters and neurons in the various layers is noted above the respective blocks.

6.5. Experimental Setup

In this section, we present and analyze the performed optimization-based meta-learning experiments. Specifically, we conducted 1-shot 2-ways, 1-shot 5-ways, 3-shots 5-ways, and 5-shots 5-ways experiments. The algorithm and methods presented are mainly analyzed in the 5-ways setup, to depict their advantages. The algorithm has been developed in the Python programming language through the TensorFlow® module. The performance tests for the state-of-the-art comparison, have been performed on a eight generation Intel® Core™ i5 processor (4-cores). At the edge side, the Raspberry® Pi4 and NCS 2 have been employed. Consequently, the RaspbianOS operating system has been utilized. To run the model on NCS 2 and optimize the inference process, we used the OpenVino module on Python.

6.5.1. Meta-learning Experiments

All experiments have been performed in a similar setup for the two topologies (CNN and Conv-VAE + Dense). For the topology with the Conv-VAE, the network on the $\mathcal{D}^{m-train}$ dataset is first pre-trained. The employed loss function and optimizer are binary crossentropy and Adam respectively. A learning rate of 1e-4 is used for Adam. The training is conducted on 200 epochs with a latent dimension of 15, i.e., 30 descriptive parameters of the set of multivariate Gaussian distributions. Since Conv-VAE is part of the category of deep generative networks, it can also partially reconstruct $\mathcal{D}^{m-test}$ instances without further training. An example of reconstruction on sampled classes from both $\mathcal{D}^{m-train}$ and $\mathcal{D}^{m-test}$ is displayed in Fig. 6.15.

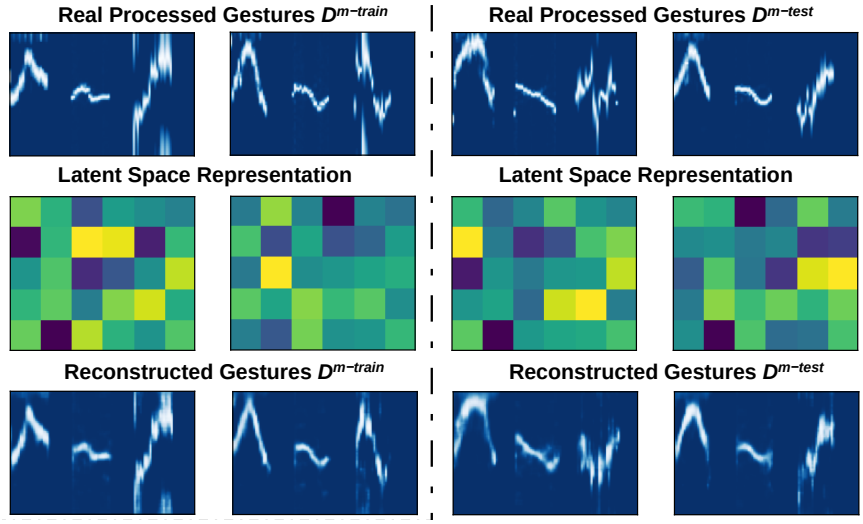


Figure 6.15: Example of latent space generation (heatmap representation) and example reconstruction using Conv-VAE. For better visualization of the instances, the RTM, DTM, and ATM channels are concatenated as a single image.

For the 5 ways experiments, the task training is performed through 4 inner epochs and an inner-batch of size 2 for 1-shot, and size 3 otherwise. For both *base-learning* and *meta-learning* phases, the Adam optimizer is used with β_1 and β_2 equal to 0 and 0.5 respectively. The inner learning rate is set to 8e-4, whereas the meta-learning rate has an initial value of 7e-4 with a decay step of 2,000. The chosen number of meta-iterations is 2,200, while the classes for each task are randomly sampled by $\mathcal{D}^{m-train}$. The loss function chosen for the classification is categorical crossentropy. In the evaluation phase, accuracy statistics are saved and processed every 220 iterations in the shape of box plots. For experiments with the EGNS, a task buffer of length 5 has been chosen, with a $\mathcal{D}^{m-train}$ validation accuracy threshold of 89 %,

95 % and 98 % for 1-shot, 3-shots, and 5-shots, respectively. For the TSGC experiments, the gradient is clipped when the L2 norm exceeds 0.5. For the DMCW, a total of 10 samples per class is used for the computation of the weights. The generated models are finally tested on 1,000 tasks sampled by $\mathcal{D}^{m-test}$. For DMCW and EGNS, the final task training is performed as a traditional single-task optimization approach. For TSGC, gradient clipping is also executed on the training batches. The EGNS and DMCW are exclusively used during meta-iterations, to increase the model's generalization capability over one or a few new examples of unseen classes. The achieved prediction accuracy, model size, adaptation time, and latency are evaluated and compared with state-of-the-art techniques. In both the evaluation and testing phases, 10 examples per class are used for testing. This means that in the 5-ways experiments, 50 test examples per task are utilized.

6.5.2. Performance Evaluation

We first present the results obtained on a single experiment, showing the benefits achievable on unseen classes thanks to an optimization-based meta-learning approach. Then, we conduct an ablation study, by analyzing the contributions of the individual proposed methods, for both proposed topologies. Next, we compare our achieved results with those of some existing techniques in terms of neural network size, prediction accuracy, and latency. All the experiments for the proposed methods and ablation study have been performed on a 4-core eight generation Intel® Core™ i5 processor. Regarding adaptation at the Edge, we display the results of adaptation time to new tasks and model deployment on Raspberry® Pi4 and NCS 2.

6.5.2.1. Experiment Analysis

The metric used to evaluate the training performance of each model is validation accuracy. This parameter is estimated after each meta-iteration, by evaluating the model on new sampled tasks. For each validation two tasks are sampled by the $\mathcal{D}^{m-train}$ and $\mathcal{D}^{m-test}$ respectively. A box-plot of task statistics is built every 220 meta-iterations. Generalization ability can be assessed by observing the variation in the box plots as meta-iterations progress. In a successful experiment, we observe the increase of the median accuracy on the sequence of box plots, as well as the reduction of the intervals of percentiles and whiskers. The trend of box plots for the experiment with EGNS for the CNN topology is shown in Fig. 6.16. The contribution of EGNS is combined with the basic MAML + MSL + DA + CA algorithm, which we term $^{+}$MAML. Another possible way of assessing the generalization capability is to observe the distribution of validation accuracy as meta-iterations increase. Usually, for the first training tasks, the accuracy tends to assume a multimodal shape due to different complexity in tasks resolution. In

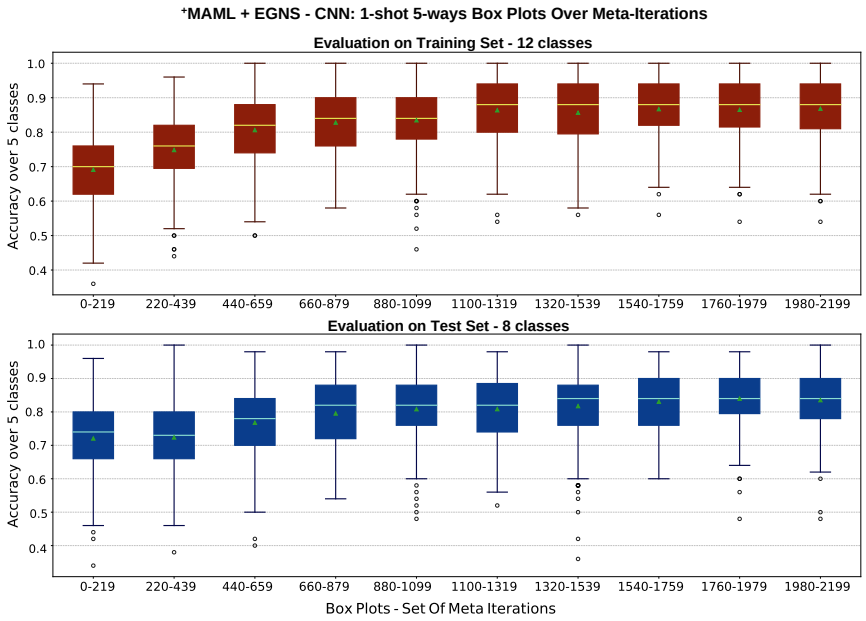


Figure 6.16: The trend of box plots generated on classification accuracy in the validation phase for the EGNS experiment with CNN topology. In red are the box plots built on the tasks sampled from the meta-train dataset, while in blue are those built over the meta-test. The mean and median values are represented for each box plot by a triangle and a line, respectively.

the training time, the model learns to resolve better new tasks thanks to the improved parametersínitialization. This leads the accuracy distribution to have a negatively skewed tendency towards the 100 % correct classification. The accuracy density histograms, generated for the first and last 220 meta-iteration box-plots, are shown in Fig. 6.17 for the CNN - EGNS experiment. The quartiles and range percentages are noted in the middle plot on a Gaussian distribution that could be associated with the box plot. Roughly by definition, 50 % of the values are contained between the first and third quartiles of the box plots. The actual accuracy distribution, however, as can be seen, does not assume a Gaussian shape.

The generalization outcome can even be observed on the individual classes by generating a cumulative confusion matrix for sets of meta-iterations. In Fig. 6.18 are depicted the confusion matrices of the first and last 550 meta-iterations for the EGNS with CNN topology experiment. As can be noticed from the matrices, as the iterations progress, the model learns to solve quicker new tasks thanks to the updated initialization. This also applies to the unseen classes belonging to $\mathcal{D}^{m-test}$.

Some actions are more complex to distinguish between each other because of similarities in patterns, thus leading to specific prediction errors. It can be

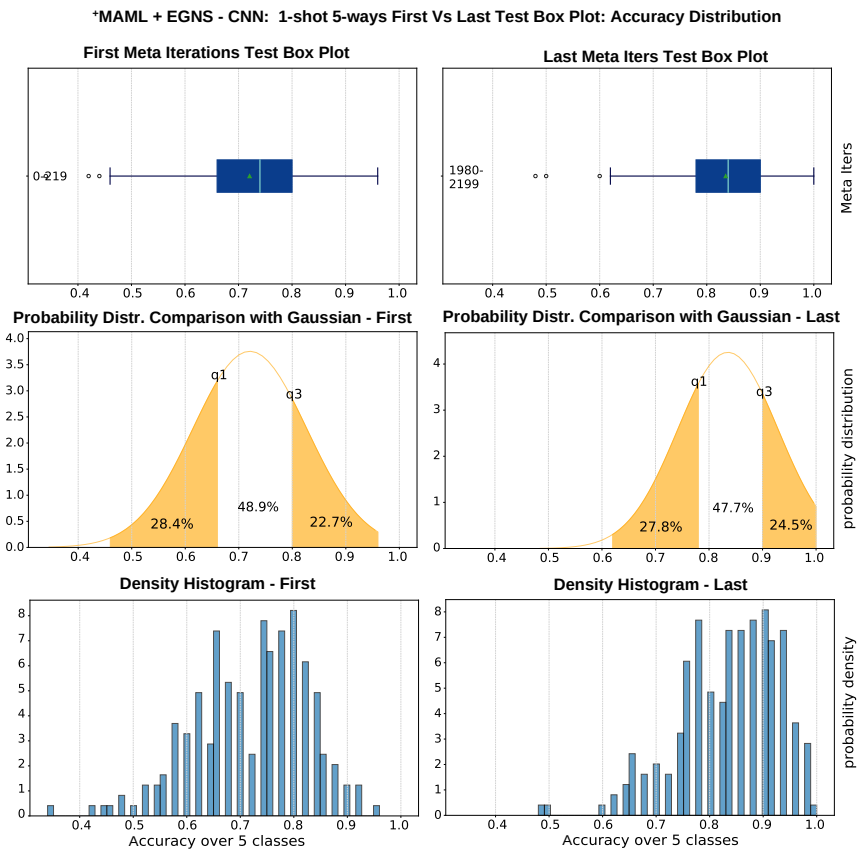


Figure 6.17: Density histogram of validation accuracy on test for the EGNS experiment with CNN topology. Values q1 and q3 on the Gaussian indicate first and third quartiles, respectively. Percentages indicate the amount of data in the sections of the distribution. The accuracy, which does not assume a Gaussian distribution, exhibits a negative skew for the last 220 meta-iterations.

noted, for example, the misclassification between right swipe and diagonal nw-se in the confusion matrix on $\mathcal{D}^{m-train}$ and specularly that between left-swipe and diagonal nw-sw for $\mathcal{D}^{m-test}$.

6.5.2.2. Results Analysis

All the experiments have been performed for both the proposed topologies, analyzing the combination of the presented methods against the base algorithm $^{+}$MAML. Each experiment, tested on 1,000 final test tasks, has been repeated three times. The average accuracy results for the 5-way experiments are presented in Table 6.2 and Table 6.3, respectively.

For the CNN topology experiments, the total number of trainable para-

Table 6.2: CNN topology. Average accuracy results of 5-ways experiments with 95 % confidence intervals, computed over 1,000 final test tasks of $\mathcal{D}^{m-test}$. Individual methods are implemented in each experiment to the base algorithm.

Accuracy 5-ways	**1-shot [%]**	**3-shots [%]**	**5-shots [%]**
Base ($^{+}$MAML)	82.85 ± 0.55	92.07 ± 0.31	**94.19 ± 0.26**
DMCW	82.32 ± 0.56	90.80 ± 0.40	93.64 ± 0.27
EGNS	**84.36 ± 0.55**	**92.54 ± 0.30**	93.87 ± 0.25
TSGC	**84.28 ± 0.56**	**92.85 ± 0.30**	94.16 ± 0.26
DMCW+EGNS	82.53 ± 0.57	91.10 ± 0.34	93.85 ± 0.25
DMCW+TSGC	82.81 ± 0.53	91.39 ± 0.34	93.49 ± 0.26
EGNS+TSGC	83.97 ± 0.54	92.02 ± 0.32	94.15 ± 0.25
All	83.29 ± 0.55	91.90 ± 0.31	93.92 ± 0.26

Table 6.3: Conv-VAE+Dense topology. Average accuracy results of 5-ways experiments with 95 % confidence intervals, computed over 1,000 final test tasks of $\mathcal{D}^{m-test}$. Individual methods are implemented in each experiment to the base algorithm.

Accuracy 5-ways	**1-shot [%]**	**3-shots [%]**	**5-shots [%]**
Base ($^{+}$MAML)	76.31 ± 0.63	87.18 ± 0.38	90.78 ± 0.31
DMCW	76.86 ± 0.61	88.95 ± 0.38	92.06 ± 0.30
EGNS	78.69 ± 0.61	87.99 ± 0.37	90.90 ± 0.29
TSGC	78.67 ± 0.60	88.24 ± 0.35	91.18 ± 0.28
DMCW+EGNS	78.67 ± 0.57	89.46 ± 0.37	92.59 ± 0.27
DMCW+TSGC	**80.09 ± 0.59**	**90.59 ± 0.33**	**92.97 ± 0.27**
EGNS+TSGC	**80.25 ± 0.58**	88.73 ± 0.34	92.07 ± 0.28
All	79.19 ± 0.57	89.93 ± 0.33	92.59 ± 0.26

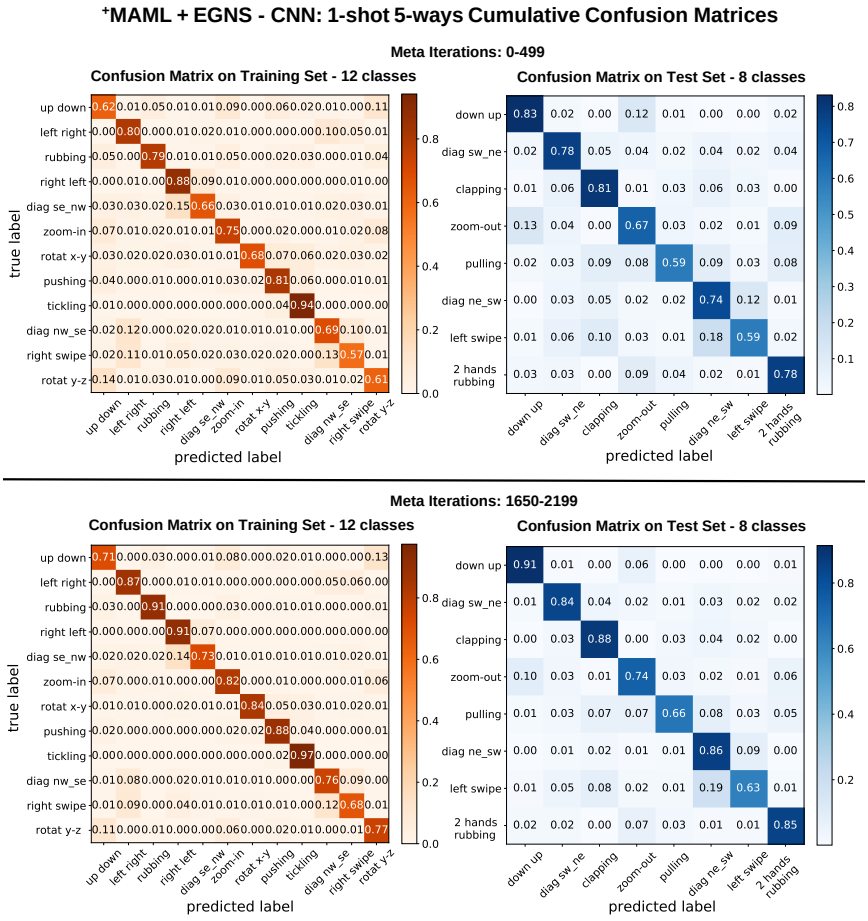


Figure 6.18: Cumulative confusion matrices for the EGNS experiment with the CNN topology. Confusion matrices are obtained on the first and last 550 meta-iterations in the validation phase for both training and test classes.

meters in the model is 1,562,629. This large number of parameters, as can be noticed through accuracy results in Table 6.2, allows the model to generalize well, guaranteeing with fast-adaptation, excellent results on unseen classes. For such a topology, the DMCW method brings no performance benefit. The model size enables extracting more features from each data while query update after each epoch through MSL reduces the possibility of overfitting. The DMCW is even more counterproductive as the number of shots increases. In such a case, the model comprehends better the differences among classes, thanks to the higher number of examples. For 1 and 3 shots experiments, individual use of EGNS and TSGC leads to the highest accuracy. These techniques give less importance to single tasks, thus favoring the meta-learning objective. In the 5-shot approach, the $^{+}$MAML algorithm, allows without additional contributions, to achieve the highest accuracy. The information

Table 6.4: Average accuracy results of 1-shot 2-ways experiments with 95 % confidence intervals, computed over 1,000 final test tasks of $\mathcal{D}^{m-test}$. Individual methods are applied in each experiment to the base algorithm, for both topologies.

Accuracy 1-shot 2-ways	**CNN [%]**	**Conv-VAE+Dense [%]**
Base ($^{+}$MAML)	90.57 ± 0.73	88.89 ± 0.77
DMCW	90.78 ± 0.71	86.71 ± 0.98
EGNS	91.17 ± 0.73	87.62 ± 0.84
TSGC	**92.70 ± 0.67**	**91.23 ± 0.69**
DMCW+EGNS	91.53 ± 0.71	86.51 ± 1.00
DMCW+TSGC	91.38 ± 0.73	90.43 ± 0.74
EGNS+TSGC	**92.88 ± 0.64**	90.86 ± 0.69
All	**93.05 ± 0.63**	87.32 ± 0.86

provided by the training instances is then enough to compensate for possible overfitting and exploding gradients. For the first topology, none of the experiments where the contributions are combined bring accuracy benefits. For a model with high feature extraction capability, the use of both EGNS and DMCW techniques can make each task locally complex and misleading. Thus, decreasing the significance of the meta-learning updates.

For simulations with the Conv-VAE+Dense topology, the total number of trainable parameters in the model drops to only 118,851. Due to input information mapping to small size, individual tasks may be more affected by overfitting phenomena. In this case, the DMCW introduces benefits compared to the basic version of the algorithm. This contribution is also beneficial with 3 and 5 shots, probably supporting the classification of the compressed information squeezed by the backbone. For this topology, the best results are achieved by combining the DMCW and TSGC methods. The classification of low-dimensional representations is aided by class weighting for individual tasks. The TSGC, on the other hand, avoids exploding gradient and gives more importance to the outer loop update at the end of each inner epoch. The combination of the three methods brings equal or less satisfactory results than combining two of them for 1 and 3 shot experiments. With the use of 5 shots, the single techniques contributions do not lead to better results than with $^{+}$MAML. This is probably due to the higher amount of data available, which leads to smoother task training. The accuracy results for both topologies in the 1-shot 2-ways approach are presented in Table 6.4.

For 1-shot 2-ways experiments, the greatest benefits are achieved through TSGC for both the topologies. For a 2-ways application, the DMCW contributions are counterproductive or not significant. Class weighting with only two categories can easily skew the learning towards one of them, especially

Table 6.5: Training times to adapt to new tasks for both topologies on the 4-core Intel® CPU. Times, given a number of ways and shots, are calculated as the average of the adaptation time of all experiments, each tested and averaged over 1,000 final test tasks of $\mathcal{D}^{m-test}$.

Adaptation Time	**CNN [ms]**	**Conv-VAE+Dense [ms]**
1-shot 2-ways	498	206
1-shot 5-ways	1,180	647
3-shots 5-ways	2,548	1,254
5-shots 5-ways	4,278	1,991

Table 6.6: Best-in-class results compared to the state of the art. Average accuracy results of the experiments with 95 % confidence intervals, computed over 1,000 final test tasks of $\mathcal{D}^{m-test}$. The various algorithms have been tested under similar evaluation conditions on the 20 gestures dataset. The proposed algorithms are marked with *. FC means Fully Connected (Dense).

Accuracy	**5-ways [%]**			**2-ways [%]**
	1-shot	**3-shot**	**5-shot**	**1-shot**
Reptile	63.95 ± 0.62	86.82 ± 0.39	89.68 ± 0.40	87.27 ± 0.84
MAML (2^{nd} Ord.)	78.95 ± 0.59	90.71 ± 0.36	93.41 ± 0.23	88.53 ± 0.78
LGM-Net	77.57 ± 0.20	84.38 ± 0.14	89.98 ± 0.10	85.04 ± 0.17
Weighting Net	81.17 ± 0.48	**93.47 ± 0.30**	**94.47 ± 0.28**	**95.61 ± 0.43**
CNN*	**84.36 ± 0.55**	92.54 ± 0.30	94.19 ± 0.26	93.05 ± 0.63
Conv-VAE+FC*	80.25 ± 0.58	90.59 ± 0.33	92.97 ± 0.27	91.23 ± 0.69

with small input sizes as for the second topology. For similar reasons, the model can learn to over-depend on noise augmented inputs via EGNS and rank worse on the test data. For CNN, the use of combined EGNS and TSGC brings some benefits, mainly preventing overfitting in the base-learning phase, given the higher simplicity of the tasks. The accuracy reached with the three techniques combined depicts how preventing over-dependence on the individual tasks can favor the generalization aim.

The average adaptation times to new tasks on the 4-core Intel® CPU for the two topologies are listed in Table 6.5.

As can be seen from the table, the model size of the Conv-VAE topology, which is an order of magnitude smaller than the CNN, allows a reduction of the adaptation time by half for the 1-shot experiments. The time required to adapt to a new task is further reduced for Conv-VAE when more than 1 example per class is employed. Regardless of the method used, the inference time on CPU to predict the class of a single example is on average 64 ms for both topologies in the 5-ways approach.

6.5.2.3. Comparison with Existing Techniques

The best-achieved results, obtained through the various experiments and topologies, are compared with both meta-learning state-of-the-art and classical optimization-based algorithms, trained on $\mathcal{D}^{m-train}$ and tested on $\mathcal{D}^{m-test}$. Respectively, the Reptile [69] and MAML algorithms for the optimization-based class and Weighting Net and LGM-Net, employed in the papers [61] and [62] are trained on our proposed gestures dataset. For the comparison, similar evaluation conditions are used. The Reptile and MAML (2nd Order) algorithms are utilized to train the proposed CNN topology for 2,200 meta-iterations. The topology presented in [61], adapted to 3-channel gesture information, has been employed for the LGM-Net. The Weighting Net, with a feature dimension of 64, has been adapted to the shape of the gestures and, the relative embedding module has been trained to extract features only from the *support* instances. The accuracy results for the state-of-the-art algorithms, averaged over three repetitions, are presented in Table 6.6.

As can be noticed from Table 6.6, the proposed method with CNN topology performs the best in the 1-shot 5-ways experiment, leading to better results than the Weighting Net by around 3 %. In all the other experiments, the proposed method performs slightly less accurately only compared to the Weighting Net. With more than one shot, the Weighting Net has the advantage of being able to mediate the predictions obtained thanks to a sequence of comparisons of the *query* image with those of *support*. However, with the availability of only one example per class, it lacks this great feature and loses robustness. The proposed methods though, lead in all the experiments to better results than all the other optimization-based methods. For simple experiments (2-ways) or a higher number of shots, the difference in accuracy obtained between the methods gets narrower. In such conditions, even the simplest algorithms can achieve high feature extraction from samples. So, the resolution of the tasks becomes less dependent on the initialization making the employed generalization techniques less effective.

The comparison in terms of model size is presented in Table 6.7.

In terms of the number of parameters, the Conv-VAE+Dense approach enables the generation of an order of magnitude smaller models compared to the CNN. Even if in terms of accuracy the second topology performs a few percentage points worse than the Weighting Net, it requires about half as many parameters for tasks resolution. Furthermore, among the compared methods, the Conv-VAE topology results in the one with the least number of required variables.

Table 6.8 presents the time required for adaptation to a new task (Ta) and the single-sample inference (Ti) for the considered algorithms. Reptile and MAML are tested using the same methodology as the proposed optimization-based models. As they are utilized on the CNN topology, they lead to results very similar to those of the proposed methods and are, therefore, excluded

Table 6.7: Best-in-class results compared to the state of the art. Number of trainable parameters per topology and experiment, computed over 1,000 final test tasks of $\mathcal{D}^{m-test}$. The various algorithms have been tested under similar evaluation conditions.

Model Size	**5-ways**	**2-ways**
Reptile	1,562,629	1,513,474
MAML (2nd Order)	1,562,629	1,513,474
LGM-Net	300,421	298,882
Weighting Net	229,157	226,034
Proposed (CNN)	1,562,629	1,513,474
Proposed (Conv-VAE+Dense)	**118,851**	**118,464**

Table 6.8: Best-in-class results compared to the state of the art. Adaptation time (Ta) and latency of prediction on single sample (Ti) per topology and experiment, computed over 1,000 final test tasks of $\mathcal{D}^{m-test}$. The various algorithms have been tested under similar evaluation conditions on the 4-core Intel® CPU. The proposed algorithms are marked with *

Adaptation (Ta) + single prediction time (Ti)	**5-ways [ms]**			**2-ways [ms]**
	1-shot	**3-shot**	**5-shot**	**1-shot**
	Ta + Ti	Ta + Ti	Ta + Ti	Ta + Ti
LGM-Net	1,710	4,310	6,290	1,340
Weighting Net	**92** + 143	**278** + 215	**476** + 287	**39** + **55**
CNN*	1,180 + **64**	2,548 + **64**	4,278 + **64**	498 + 61
Conv-VAE+Dense*	647 + 64	1,254 + 64	1,991 + 64	206 + 61

from this table. For LGM-Net, the two values of Ta and Ti are summed, given the high degree of interdependence between the modules (Embedding, MetaNet, and TargetNet) in its structure. For the Weighting Net, Ta is estimated as the required time to map the *support* examples to a reduced size via the EmbeddingNet. In this case, Ti is computed as the needed time to process and classify a *query* example through the entire model pipeline after the *support* adaptation. In terms of adaptation time (Ta), the proposed models take longer than the Weighting Net. On the other hand, the optimization-based models enable the instance classification in a significantly short time (Ti) and in a way that is independent of the number of training shots. In the 5-shot experiment, the proposed topologies require only a quarter of the time needed by the Weighting Net for prediction. This brings a huge advantage in real-time applications or implementations at the edge.

Table 6.9: Training times to adapt to new tasks for both topologies on Raspberry® Pi4 (without NCS 2). Times, given a number of ways and shots, are calculated as the average of the adaptation time of all experiments, each tested and averaged over 10 final test tasks of $\mathcal{D}^{m-test}$.

Adaptation Time	CNN [ms]	Conv-VAE+Dense [ms]
1-shot 2-ways	3,655	983
1-shot 5-ways	7,976	2,572
3-shots 5-ways	22,188	5,851
5-shots 5-ways	37,310	9,416

Table 6.10: Training times to adapt to new tasks for both topologies on Raspberry® Pi4 plus deployment time on NCS 2. Times, given a number of ways and shots, are calculated as the average of the adaptation time of all experiments, each tested and averaged over 10 final test tasks of $\mathcal{D}^{m-test}$.

Adaptation Time	CNN [ms]	Conv-VAE+Dense [ms]
1-shot 2-ways	3,678	2,265
1-shot 5-ways	7,416	3,544
3-shots 5-ways	13,283	4,562
5-shots 5-ways	21,567	6,701

6.5.2.4. Edge Implementation

The topologies presented in this paper use only NCS 2 compatible layers and procedures. All models, pre-trained with the optimization-based approach on the 4-cores CPU, are adapted at the edge to single tasks generated by $\mathcal{D}^{m-test}$ via Raspberry® Pi4. The models are first tested on the Raspberry® Pi4 without connecting NCS 2, consequently using only the 4 ARM cores, to estimate adaptation time and inference on single sample. The computed task adaptation time are presented in Table 6.9. The models are then deployed on the NCS 2 and, prediction inference for each test sample is conducted at the device level. For the various experiments, the achieved results in terms of summation of task adaptation time on Raspberry® Pi4 and deployment on NCS 2 are presented in Table 6.10. As can be noticed from the Table 6.9 and Table 6.10, as the number of samples per class increases, the time to adapt to a new task rises significantly for the CNN topology, requiring up to more than 21 s for an adaptation and deployment on the NCS 2. On the contrary, the Conv-VAE+Dense, given the much smaller number of parameters, requires less than 7 s for a 5-shots task. Conv-VAE+Dense can therefore lead to a saving of up to about two-thirds of the time. The results from Table 6.9 highlight generally longer adaptation times for both

topologies on the Raspberry®. This is mainly due to computation limits, especially for the CNN topology as the number of shots increases. In addition, the models, once deployed on the NCS 2, allow much shorter single inference times (Ti) and therefore enable potential real-time applications with very low latency. The needed time for a single prediction after model adaptation is topology-dependent. For both 2-ways and 5-ways experiments, the model on NCS 2 requires an average of 5 ms and 4 ms for CNN and Conv-VAE+Dense, respectively. These values are significantly lower than those obtained only via Raspberry® Pi4 (Table 6.9), where the prediction of a single example takes on average 351 and 333 ms for CNN and Conv-VAE+Dense, respectively. The demonstrated results underscore how deploying on the NCS 2 can be a very advantageous strategy when very low latency is required. Since the adaptation is performed offline, the single inference time does not consider the time required for gesture sampling (Ts) and preprocessing time (Tp). These times are dependent on the type of gesture performed, its intrinsic duration, and the number of recorded frames before applying zero padding. Table 6.11 presents the computed Ts and Tp times for one example per class of each of the 20 gestures. The Tp values are obtained over an average of 10 preprocessing repetitions of the same example performed on Raspberry® Pi4. Thus, the total (end-to-end) time consists of the sum of Ts+Tp+Ti.

6.6. Conclusion

In this paper we present a complete pipeline based on hand gestures performed on an FMCW radar, to exhibit a proof-of-concept of user-adaptability for novel unseen hand poses. The system solution, based on data collected for twenty different types of gestures, from five users in three different environments, allows not only the extraction of useful features of performed actions but also a fast adaptation to new gestures. The pipeline is composed of a first preprocessing phase, then a meta-learning approach to generate the best possible model initialization, and an edge-suitable adaptation to new tasks and classes never faced in the training phase. The specific preprocessing employed, thanks to the combination of techniques both in the frequency and time domain, allows extracting the main information of the gestures only, thus significantly reducing the size of the raw data collected by radar. The information constructed for each gesture, in the form of 3 channels, represents the hand distance from the radar, the action velocity, and the azimuth angle of arrival. A meta-learning optimization-based approach, trained on twelve of the processed gestures, depicts how new never faced tasks can be more easily solved, thanks to the context information extracted in the training phase. Three techniques, aiming at increasing the generalization ability of the model in comparison to the state-of-the-art, are presented: dynamic meta-class weighting, task-specific gradient clipping, and evaluation-

Table 6.11: Recording (Ts) and preprocessing (Tp) times computed for a random example belonging to each class of gestures. The time Tp is computed over an average of 10 preprocessing repetitions on Raspberry® Pi4.

Gesture	Ts [ms]	Tp [ms]
Down - Up	2,700	1,198
Up - Down	2,900	1,285
Left - Right	400	209
Rubbing	2,900	1,320
Right - Left	1,800	802
Diagonal SW-NE	1,900	884
Diagonal SW-NW	2,600	1,173
Clapping	1,700	786
Zoom-In	2,000	905
Zoom-Out	2,100	942
X-Y Rotation	3,100	1,370
Pushing	3,100	1,399
Pulling	2,500	1,116
Tickling	3,100	1,408
Diagonal NW-SE	1,600	719
Diagonal NW-SW	2,100	941
Right Swipe	1,600	721
Left Swipe	2,000	890
Y-Z Rotation	1,800	815
Two Hands Rubbing	2,600	1,166

based Gaussian noise summation respectively. The introduced methods have the great advantage of improving the model's parameters initialization in the training phase without directly affecting the final adaptation setup on the eight test classes. This enables both a more versatile implementation at the edge and a very fast prediction on new samples, reducing remarkably the computation latency. Further, compared to other state-of-the-art techniques, the optimization-based approach doesn't involve the comparison of the query samples with the support ones in the test phase, thus, bringing to an additional time latency reduction. Two different topologies for task resolution are presented. A first topology based on a series of convolution layers consents feature extraction for each sample thanks to a large number of defined parameters. A second topology instead, employs the encoding part of a Conv-VAE as a backbone to efficiently extract features, thus greatly reducing the number of model parameters. For such a topology, a greater effect of the presented optimization techniques is visible, thanks to the various contributions that counteract the effects of overfitting, exploding gradient, and meta-overfitting. Thanks to these features, this topology enables the generation of models that perform very well in terms of accuracy but with half the variables required in comparison to state-of-the-art. Moreover, the results obtained at the edge optimistically show how these algorithms can be used for real-time applications, aiding the adaptation to new users, gestures, and situations. To the best of our knowledge, this is the first user-adaptable model implemented at the edge for radar-based HCI.

On the other hand, the generated models lead to an accuracy that is lower than the state-of-the-art in several experiments. Other meta-learning algorithms, based on the classification of relations among examples, have the inherent advantage of leading to more robust predictions. Future work will explore the application at the edge of relational algorithms and potential methods of reducing the model size without harming generalization capabilities. Experiments with a broader set of gestures and examples will also be conducted, examining the generalization ability of the models across various splits of users and environments. Adaptive interfacing, based on an approach such as the one presented in this paper may be exploitable for people with motor or visual impairments, due to their inability to perform classic actions in a conventional manner. Since individuals without disabilities have been considered in this current work, direct studies will be conducted on analyzing how much radar adaptive interfacing can be used and relied on in these particular use cases.

6.7. Acknowledgments & Declarations

- **Funding.** The work presented is partially supported by the ITEA3 UPSIM project (N°19006) funded by the German Federal Ministry of

Education and Research (BMBF), the Austrian Research Promotion Agency (FFG), the Rijksdienst voor Ondernemend Nederland (Rvo) and the Innovation Fund Denmark (IFD).

References

[1] Alan Dix, Janet Finlay, Gregory D Abowd, and Russell Beale. *Human-computer interaction*. Pearson Education, 2003.

[2] Karan Ahuja, Paul Streli, and Christian Holz. Touchpose: Hand pose prediction, depth estimation, and touch classification from capacitive images. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 997–1009, 2021.

[3] Jangwoon Kim, Jaewan Park, HyungKwan Kim, and Chilwoo Lee. Hci (human computer interaction) using multi-touch tabletop display. In *2007 IEEE Pacific Rim conference on communications, computers and signal processing*, pages 391–394. IEEE, 2007.

[4] Sridher Kaminani. Human computer interaction issues with touch screen interfaces in the flight deck. In *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, pages 6B4–1. IEEE, 2011.

[5] Julie A Jacko. *Human computer interaction handbook: Fundamentals, evolving technologies, and emerging applications*. CRC press, 2012.

[6] Piercarlo Dondi, Luca Lombardi, and Marco Porta. Human-computer interaction through time-of-flight and rgb cameras. In *International Conference on Image Analysis and Processing*, pages 89–98. Springer, 2011.

[7] Yiwei Wang and Cheolkon Jung. Interaction-free hand segmentation using kinect camera. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 4593–4593. IEEE, 2017.

[8] Yunxin Duan, Hui Deng, and Feng Wang. Depth camera in human-computer interaction: An overview. In *2012 Fifth International Conference on Intelligent Networks and Intelligent Systems*, pages 25–28. IEEE, 2012.

[9] Martin Böhme, Martin Haker, Thomas Martinetz, and Erhardt Barth. A facial feature tracker for human-computer interaction based on 3d tof cameras. *Int. J. on Intell. Systems Techn. and App., Issue on Dynamic 3D Imaging*, 5(3):4, 2008.

[10] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7356–7365, 2018.

[11] Zhangjie Fu, Jiashuang Xu, Zhuangdi Zhu, Alex X Liu, and Xingming Sun. Writing in the air with wifi signals for virtual reality devices. *IEEE Transactions on Mobile Computing*, 18(2):473–484, 2018.

[12] Huan Yan, Yong Zhang, Yujie Wang, and Kangle Xu. Wiact: A passive wifi-based human activity recognition system. *IEEE Sensors Journal*, 20(1):296–305, 2019.

[13] Lili Chen, Xiaojiang Chen, Ligang Ni, Yao Peng, and Dingyi Fang. Human behavior recognition using wi-fi csi: Challenges and opportunities. *IEEE Communications Magazine*, 55(10):112–117, 2017.

[14] Hui-Shyong Yeo and Aaron Quigley. Radar sensing in human-computer interaction. *interactions*, 25(1):70–73, 2017.

[15] Bo Li, Xiaotian Yu, Fan Li, and Qiming Guo. Deep learning based target activity recognition using fmcw radar. In *International Conference on Artificial Intelligence for Communications and Networks*, pages 484–490. Springer, 2020.

[16] Prachi Vaishnav and Avik Santra. Continuous human activity classification with unscented kalman filter tracking using fmcw radar. *IEEE Sensors Letters*, 4(5):1–4, 2020.

[17] Aashni Haria, Archanasri Subramanian, Nivedhitha Asokkumar, Shristi Poddar, and Jyothi S Nayak. Hand gesture recognition for human computer interaction. *Procedia computer science*, 115:367–374, 2017.

[18] Yanan Xu and Yunhai Dai. Review of hand gesture recognition study and application. *Contemporary Engineering Sciences*, 10(8):375–384, 2017.

[19] Jaime Lien, Nicholas Gillian, M Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. Soli: Ubiquitous gesture sensing with millimeter wave radar. *ACM Transactions on Graphics (TOG)*, 35(4):1–19, 2016.

[20] Munir Oudah, Ali Al-Naji, and Javaan Chahl. Hand gesture recognition based on computer vision: a review of techniques. *journal of Imaging*, 6(8):73, 2020.

[21] Shahzad Ahmed, Karam Dad Kallu, Sarfaraz Ahmed, and Sung Ho Cho. Hand gestures recognition using radar sensors for human-computer-interaction: A review. *Remote Sensing*, 13(3):527, 2021.

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[24] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[25] Mário P Véstias. Deep learning on edge: Challenges and trends. *Smart Systems Design, Applications, and Challenges*, pages 23–42, 2020.

[26] Marian Verhelst and Bert Moons. Embedded deep neural network processing: Algorithmic and processor techniques bring deep learning to iot and edge devices. *IEEE Solid-State Circuits Magazine*, 9(4):55–65, 2017.

[27] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and<0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[28] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[29] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021.

[30] Sridhar Swaminathan, Deepak Garg, Rajkumar Kannan, and Frederic Andres. Sparse low rank factorization for deep neural network compression. *Neurocomputing*, 398:185–196, 2020.

[31] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[32] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *arXiv preprint arXiv:2003.03033*, 2020.

[33] He Li, Kaoru Ota, and Mianxiong Dong. Learning iot in edge: Deep learning for the internet of things with edge computing. *IEEE network*, 32(1):96–101, 2018.

[34] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[36] Gary Marcus. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*, 2018.

[37] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 0210–0215. IEEE, 2018.

[38] Mohammad M Masud, Clay Woolam, Jing Gao, Latifur Khan, Jiawei Han, Kevin W Hamlen, and Nikunj C Oza. Facing the reality of data stream classification: coping with scarcity of labeled data. *Knowledge and information systems*, 33(1):213–244, 2012.

[39] Joaquin Vanschoren. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548*, 2018.

[40] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

[41] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

[42] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR, 2016.

[43] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International conference on learning representations*, 2016.

[44] Saverio Trotta, Dave Weber, Reinhard W Jungmaier, Ashutosh Baheti, Jaime Lien, Dennis Noppeney, Maryam Tabesh, Christoph Rumpler, Michael Aichner, Siegfried Albel, et al. Soli: A tiny device for a new human machine interface. In *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, volume 64, pages 42–44. IEEE, 2021.

[45] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[46] Siddharth S Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial intelligence review*, 43(1):1–54, 2015.

[47] K Martin Sagayam and D Jude Hemanth. Abc algorithm based optimization of 1-d hidden markov model for hand gesture recognition applications. *Computers in Industry*, 99:313–323, 2018.

[48] Quentin De Smedt, Hazem Wannous, and Jean-Philippe Vandeborre. Heterogeneous hand gesture recognition using 3d dynamic skeletal data. *Computer Vision and Image Understanding*, 181:60–72, 2019.

[49] Bo Liao, Jing Li, Zhaojie Ju, and Gaoxiang Ouyang. Hand gesture recognition with generalized hough transform and dc-cnn using realsense. In *2018 Eighth International Conference on Information Science and Technology (ICIST)*, pages 84–90. IEEE, 2018.

[50] Dinh-Son Tran, Ngoc-Huynh Ho, Hyung-Jeong Yang, Eu-Tteum Baek, Soo-Hyung Kim, and Gueesang Lee. Real-time hand gesture spotting and recognition using rgb-d camera and 3d convolutional neural network. *Applied Sciences*, 10(2):722, 2020.

[51] Reza Azad, Maryam Asadi-Aghbolaghi, Shohreh Kasaei, and Sergio Escalera. Dynamic 3d hand gesture recognition by learning weighted depth motion maps. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(6):1729–1740, 2018.

[52] Amit Das, Ivan Tashev, and Shoaib Mohammed. Ultrasound based gesture recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 406–410. IEEE, 2017.

[53] Yue Zheng, Yi Zhang, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. Zero-effort cross-domain gesture recognition with wi-fi. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pages 313–325, 2019.

[54] Sruthy Skaria, Akram Al-Hourani, Margaret Lech, and Robin J Evans. Hand-gesture recognition using two-antenna doppler radar with deep convolutional neural networks. *IEEE Sensors Journal*, 19(8):3041–3048, 2019.

[55] Hyo Ryun Lee, Jihun Park, and Young-Joo Suh. Improving classification accuracy of hand gesture recognition based on 60 ghz fmcw radar with deep learning domain adaptation. *Electronics*, 9(12):2140, 2020.

[56] Mateusz Chmurski, Gianfranco Mauro, Avik Santra, Mariusz Zubert, and Gökberk Dagasan. Highly-optimized radar-based gesture recognition system with depthwise expansion module. *Sensors*, 21(21):7298, 2021.

[57] Elahe Rahimian, Soheil Zabihi, Amir Asif, Dario Farina, Seyed Farokh Atashzar, and Arash Mohammadi. Fs-hgr: Few-shot learning for hand gesture recognition via electromyography. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2021.

[58] Zhi Lu, Shiyin Qin, Xiaojie Li, Lianwei Li, and Dinghao Zhang. One-shot learning hand gesture recognition based on modified 3d convolutional neural networks. *Machine Vision and Applications*, 30(7):1157–1180, 2019.

[59] Naveen Madapana and Juan P Wachs. Hard zero shot learning for gesture recognition. In *2018 24th international conference on pattern recognition (ICPR)*, pages 3574–3579. IEEE, 2018.

[60] Zhongyu Fan, Haifeng Zheng, and Xinxin Feng. A meta-learning-based approach for hand gesture recognition using fmcw radar. In *2020 International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 522–527. IEEE, 2020.

[61] Huaiyu Li, Weiming Dong, Xing Mei, Chongyang Ma, Feiyue Huang, and Bao-Gang Hu. Lgm-net: Learning to generate matching networks for few-shot learning. In *International conference on machine learning*, pages 3825–3834. PMLR, 2019.

[62] Xianglong Zeng, Chaoyang Wu, and Wen-Bin Ye. User-definable dynamic hand gesture recognition based on doppler radar and few-shot learning. *IEEE Sensors Journal*, 21(20):23224–23233, 2021.

[63] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[64] Gianfranco Mauro, Mateusz Chmurski, Muhammad Arsalan, Mariusz Zubert, and Vadim Issakov. One-shot meta-learning for radar-based gesture sequences recognition. In *International Conference on Artificial Neural Networks*, pages 500–511. Springer, 2021.

[65] Victor C Chen. *The micro-Doppler effect in radar*. Artech House, 2019.

[66] Raymond J Weber and Yikun Huang. Analysis for capon and music doa estimation algorithms. In *2009 IEEE Antennas and Propagation Society International Symposium*, pages 1–4. IEEE, 2009.

[67] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. *arXiv preprint arXiv:1810.09502*, 2018.

[68] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR, 2013.

[69] Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

Chapter 7

Few-shot User-adaptable Radar-based Breath Signal Sensing

Gianfranco Mauro[1,2], Maria De Carlos Diez[1], Julius Ott[1,3], Lorenzo Servadei[1,3], Manuel P. Cuellar[4], Diego P. Morales[2].

1. Infineon Technologies AG, Am Campeon 1-15, 85579 Neubiberg, Germany
2. Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain
3. Department of Electrical and Computer Engineering, Technical University of Munich, Arcisstrasse 21, 80333 Munich, Germany
4. Department of Electronics and Computer Technology, University of Granada, 18071 Granada, Spain

MDPI Sensors, Volume: Sensors 23, no. 2: 804, MDPI.

- Received December 2022, Accepted January 2023, Published January 2023
- DOI: 10.3390/s23020804
- Impact factor (2022): 3.9
- JCR Rank (2022): 100/275 in category Engineering, Electrical & Electronic (Q2)

Abstract. Vital signs estimation provides valuable information about an individual's overall health status. Gathering such information usually requires wearable devices or privacy-invasive settings. In this work, we propose a radar-based user-adaptable solution for respiratory signal prediction while sitting at an office desk. Such an approach leads to a contact-free, privacy-friendly, and easily adaptable system with little reference training data. Data from 24 subjects are preprocessed to extract respiration information using a 60 GHz frequency-modulated continuous wave radar. With few training examples, episodic optimization-based learning allows for generalization to new individuals. Episodically, a convolutional variational autoencoder learns how to map the processed radar data to a reference signal, generating a constrained latent space to the central respiration frequency. Moreover, autocorrelation over recorded radar data time assesses the information corruption due to subject motions. The model learning procedure and breathing prediction are adjusted by exploiting the motion corruption level. Thanks to the episodic acquired knowledge, the model requires an adaptation time of less than one and two seconds for one to five training examples, respectively. The suggested approach represents a novel, quickly adaptable, non-contact alternative for office settings with little user motion.

Keywords: Vital sign sensing; Respiration signal; Artificial neural networks; Meta-learning; Radar; FMCW; Few-shot learning; Autocorrelation; Variational autoencoder; Signal processing.

7.1. Introduction

Estimating a person's vital parameters has always been an important research topic, as it allows tracking of health status and preventing some diseases and potential accidents [1, 2]. Vital signs include the breath wave, heartbeat, body temperature, and blood pressure. The main focus of research is the heart wave, which gives direct information about how a person's heart is working and can help prevent life-threatening events such as a heart attack or arrhythmia. The breath signal can instead provide information on how the lungs are behaving. The breath wave shape can highlight if the subject is undergoing a hyperventilation episode or if the airways are obstructed due to an allergic reaction or a physical blockage [3, 4]. The estimation of vital parameters is usually performed to diagnose a health problem caused by some often acute symptoms. On the other hand, continuous vital sign monitoring could predict and prevent the worsening of respiratory and cardiovascular diseases, which account for 32 % of worldwide deaths per year [5]. Vital parameter estimation can be performed over days with portable ambulatory devices such as the Holter monitor for electrocardiogram (ECG). For long-term measurements intended for prevention, however, ambulatory machines are not versatile due to cost, maintenance, and the limitations of

user activities. To counter this, wearable devices capable of monitoring multiple vital parameters at the same time and reporting abnormalities have emerged over the years [6, 7]. Many wearable devices, such as smartwatches, proved to help predict vital anomalies, but they also have the intrinsic need to be continuously worn. This can be hard in the case, for example, of bulkier devices for breathing sense, which can be worn by newborns or elders. Many solutions are therefore moving toward non-contact sensing techniques [8].

Some non-contact solutions employ camera sensors. Through video signal processing, it is indeed possible to extract parameters such as heart rate (HR) and respiration rate (RR), which can be particularly useful in clinical or telehealth consultations [9, 10]. The use of camera sensors, however, can be inadequate in many applications, leading, especially in long-term monitoring, to serious privacy concerns. The use of thermal sensors can be employed to partially overcome this problem [11, 12]. Yet, thermal measurements are sensitive to heat and weather conditions and are not employable in all contexts. For contact-free sensing of vital parameters, ultrasound systems can be an excellent privacy-friendly solution [13]. High-frequency systems such as radar or Wi-Fi may have additional advantages, such as a much greater spatial range and the ability to pass through surfaces [14, 15, 16]. WiFi-based solutions can be very accurate in estimating vital signs [17, 18] but often require systems with transmitting (Tx) and receiving (Rx) antennas placed in separate devices, contributing to higher power consumption than radar. Remarkable among the various radar modulations is the frequency modulated continuous wave (FMCW), which enables simultaneous estimation of the relative range, velocity, and angle of arrival of targets placed in the sensor's field of view (FoV) [19, 20]. The ability to sense static components, thanks to the frequency modulation of chirp signals sent from the Tx channels of the FCMW radar, can enable privacy-friendly tracking of targets in the FoV. Further, thanks to the micro-Doppler effect, radar can also sense small and periodic displacements generated as vital signs [21]. The collected information, preprocessed in phase, is particularly corruptible by continuous user movement. Nevertheless, non-contact vital parameter estimation can be employed in relatively static settings, such as an office, even for multi-person sensing [22].

Raw radar data are inherently difficult to interpret and often require artificial intelligence (AI) techniques to filter useful information rather than pure signal processing or computer vision. Many state-of-the-art solutions use Kalman filters to reduce measured noise in vital signs or to update the specific parameter band-pass filter limits for estimates and uncertainties of chosen state variables [23, 24, 25]. However, given the Kalman filter assumptions, it is necessary to selectively filter out corrupted data caused by random user movements to avoid corrupting subsequent vital sign estimates [23]. Other solutions employ machine learning (ML) approaches to predict vital parame-

ters in the form of time series [26] or to extract relevant information, such as arrhythmia detection [27]. In other cases, the interest is more in estimating the number of peaks in time than in reconstructing the vital signs. In [28], to decrease prediction latency, the solution uses an artificial neural network (ANN) exclusively to predict the presence of heart peaks from raw radar data using labeled ECGs. Although all AI-listed solutions enable accurate reconstructions of vital parameters or estimation of target variables, they all require a large dataset of training data on various subjects for such an achievement. While tested in various contexts and for new users, many solutions may not easily fit users with unconventional vital signs or in new recording positions and angles. To address the big data need, a specific branch of ML called Meta-Learning (Meta-L) is gaining momentum [29]. In Meta-L, the goal is context generalization. An algorithm learns to solve various tasks via episodes, leveraging the accumulated experience and only a little new available data to adapt faster in new contexts. Usually, in each episode, the model tries to address an *N–ways* task, where N represents the number of classes (one per regression) and *k–shots*, or k examples per class. Generally, tasks are sampled from the same domain, and episodic learning often occurs in two phases. In an inner step, the learning algorithm learns to solve a given task by exploiting support (S) examples. In an outer step, the ability to generalize to new tasks is estimated and tuned using query (Q) examples. In optimization-based algorithms, such as model-agnostic meta-learning (MAML) [30], the learning algorithm parameters update is performed using gradient descent in both episodic steps.

In this paper, we present a user-adaptable FMCW radar solution based on signal processing and Meta-L for breath signal estimation. The approach was tested in an office desk-workplace with a single person in the FoV. The ideal user-radar board distance is up to 40 cm. This non-contact solution is employable when the user under test performs some actions characterized by little movements, such as laughing, talking, or using the keyboard, but leads to better performances in idle scenarios. Continuous detection of the user-radar range enables user tracking and radar preprocessing adaptation. Through prior information acquired via Meta-L, the algorithm is adaptable to a new person with a single or few training examples. The episodic training is designed to extract the breath information from the radar data while minimizing the contribution of the detected motion corruption due to the user's actions. Data are collected for short sessions at the desk-workplace, from 24 different users at 2 ranges of distances, up to 30 cm and 40 cm. Among all the users, 14 are selected for training and 10 for testing. Radar data are gathered via the FMCW 60 GHz radar system with 1 Tx and 3 Rx, while a breath-sensing belt is used as a reference. In a single 30-second session, the collected radar data are first preprocessed in frequency to extract the range information of a single target user. The phase signal, which contains the

breath information, is then unwrapped for a selected set of range bins, which are dynamically adjusted over the sessions. A multi-output ANN trained episodically in 1–, 5–, and 10–shots aims to predict the user's breath signal from the phase signal. The ANN maps, via a convolutional variational autoencoder (C-VAE), the radar phase to the belt reference signal, constraining the generation of latent space to the central frequency (Fc) of breath. The overall topology scheme is depicted in Figure 7.1. A series of two-band-pass digital biquad filters selectively filter the breath signal information according to the predicted Fc. The autocorrelation of the extracted signal, performed with a sliding window, allows estimating per session the level of corruption due to user motion. The corruption information is thus employed in both episodic training and prediction to improve ANN performance by fetching the most valuable information available in motion-corrupted sessions. The main contributions of this paper are as follows:

1. Implementation, to the best of our knowledge, of the first few-shot user-adaptable radar-based breath signal sensing solution.
2. Development of a specialized radar data preprocessing pipeline that dynamically tracks the user's position relative to the board.
3. Design of a cost function that constrains the generation of the latent space of a C-VAE to the respiration Fc in a multi-output ANN.
4. Development of a corruption-based sample weighting approach that guides the breathing signal estimation in the presence of user motion.

7.2. Related Works

In this section, we first investigate methods for non-contact estimation of the breath signal, focusing mainly on high-frequency solutions that are AI-oriented. We then discuss vital sign-sensing approaches that employ Meta-L techniques.

Alizadeh et al. [31] used a 77 GHz FMCW radar to extract vital parameters from a patient lying down on a bed in a non-contact vital sign sensing solution. In this work, vital signs are estimated by purely signal-processing-based methods. An initial fast Fourier transform (FFT) is performed to extract the range information of the subject from the radar board. The Fc of the vital signs is estimated on the unwrapped phase signal downstream of a second FFT that calculates the vibrations, leading to the generation of a range-vibration map. The vital signs are then extracted via a band-pass filter. This method allows reconstruction of breath rate 94 % similar to a reference signal but places the major constraint that the only non-stationary features in the range-vibration map are the biological activities. This makes

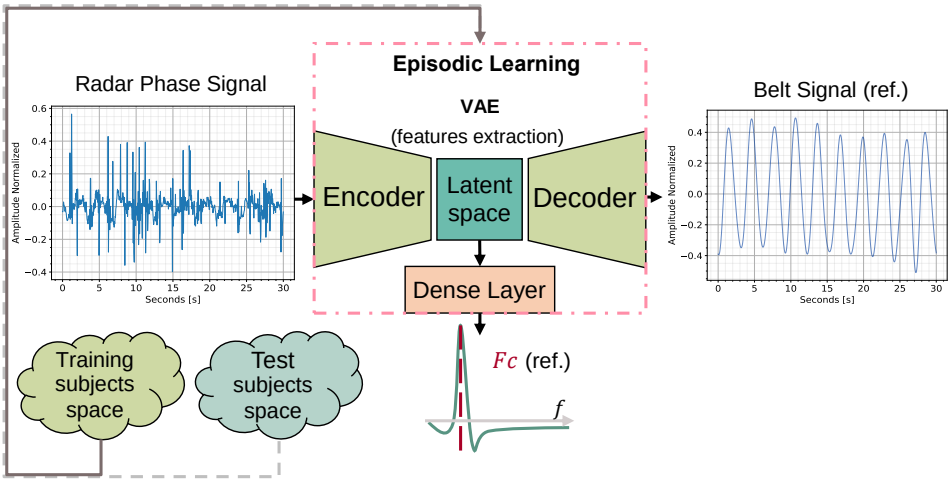


Figure 7.1: For each learning episode, a training subject is randomly sampled. For each training shot, the radar phase information is mapped to the reference belt signal (ref.) via a C-VAE. Through a dense layer, the ANN also tries to regress the extracted respiration Fc, learning from the ideal belt Fc. The latent space mapping is thus constrained to the Fc, whose estimate is also used in the prediction phase.

the solution only applicable when the subject under test is idle. Wang Y. et al. [32] proposed two different methods of vital signal estimation from phase information extracted from data collected with a 77 GHz FMCW. These methods, namely the Compressive Sensing based on orthogonal matching pursuit (CS-OMP) algorithm and the Rigrsure Adaptive soft threshold noise reduction based on discrete wavelet transform (RA-DWT), separate and reconstruct breathing and heartbeat signals instead of the more traditional band-pass filtering. Although the results obtained are very similar to those obtained with contact-based reference sensors, there is still the inherent constraint that the subject has to remain stationary in front of the radar system. Iyer et al. [27] developed a solution that uses Fourier series analysis on data collected by a 77 GHz FMCW radar to extract the vital signs of an individual from various orientations. Although the paper mainly focuses on the heartbeat for detecting arrhythmias using an ANN, the breath rate (BR) and the breathing wave are also estimated. The latter is obtained through a digital biquad band-pass filter whose parameters are invariant to the user or recording session. A filter that is not selective enough can lead to noisy predictions with many falsely detected breath peaks due to motion. Lee et al. [33] implemented a solution that detects the vital parameters of multiple subjects in the FoV using a 24 GHz FMCW Doppler radar. Doppler phase information is combined with range measurements obtained by parametric spectral estimation to distinguish multiple targets even beyond the theoreti-

cal range resolution limit. Likewise, in this approach, a band-pass filter with relatively wide bandwidth is utilized, which may not be adequate in all contexts. Lv et al. [34] used a much higher frequency FMCW 120 GHz radar system to estimate the vital signs of eight volunteers. The solution mainly focuses on acquiring the heartbeat signal, utilizing a notch filter to filter out the respiratory harmonics in the spectrum of interest. This is mainly conducted to overcome the problem of overlapping and interference of breathing and heart harmonics in some measurements. As also mentioned by the authors, the classical FFT approach does not guarantee the correct prediction of vital parameters in motion-corrupted scenarios. Gong et al. [35] illustrate an FMCW-based solution for vital sign estimation that also seeks to address the problem of sensing even in the presence of motion. The approach combines direct FMCW sensing for static instances with an indirect vital sign prediction based on motion power estimation. Two sub-long short-term memories (sub-LSTMs) are used to estimate the RR; they first classify the motion patterns and then estimate the RR. The method is robust even with some random movement patterns, such as lifting an arm, in new environments, and with new users. However, the variation in RR is estimated and not the respiratory signal, which can give additional information about a user's health quality. It is also not specified whether the users were allowed to speak during the recordings or whether this activity was taken into account. Wang D. et al. [36] proposed an interesting comparison in vital sign estimation between impulse radio ultra-wideband (IR-UWB) and FMCW radar. While radar FMCW needs phase information to extract vital signs, IR-UWB uses distance information. The data of both radar topologies are processed with a relatively standard approach that employs a band-pass filter. The IR-UWB achieves a better estimate and signal-to-noise ratio (SNR) but needs to send many pulses to distinguish the signal from noise. A high pulse rate per second also requires a high-speed analog-to-digital converter (ADC) which increases cost and hardware design complexity compared to the FMCW. For the FMCW, on the other hand, a narrow instantaneous bandwidth allows the use of lower-speed ADCs. In addition, multiple-input-multiple-output (MIMO) topologies for FMCWs allow multiple target locations and real-time monitoring. Rana et al. [37] presented a system that processes, via short-term Fourier transformation (STFT), the data collected from a UWB radar to extract vital signs. Data are collected in various areas of the house. The UWB recordings are complemented by a multi-class support vector machine (MC-SVM) that distinguishes vital signs when different activities are performed in the available locations. This approach shows preliminary results of how it is possible to recognize specific user activities with little training data. This could also potentially be used to improve the estimation of activity-related vital signs. Khan et al. [38] illustrated a channel state information (CSI) based WiFi sensing solution to track the vital signs of a patient. With the features extracted from the collected data, the health status of patients is

estimated through four types of ML algorithms via classification. These algorithms are K-nearest neighbor (KNN), decision tree, random forest, and support vector machine (SVM). The presented feature extraction approaches make it possible to preserve valid information by decreasing the dimension of the individual examples collected and simplifying the task of ML algorithms.

As far as we know, there is only one source for video-based physiological measurement that uses Meta-L and few-shot learning to estimate vital parameters. Liu et al. [39] proposed a Meta-L-based approach for personalized video-based non-contact cardiac pulse and heart rate monitoring. Thanks to the episodic training of MAML, the approach requires only 18 seconds of video for customization to new scenarios with different users, sunlight, and indoor illuminations. The solution, evaluated in two benchmark datasets, yielded substantially superior performances compared to state-of-the-art approaches.

7.3. System Description and Implementation

This section gives a general overview of the system, a description of how the data acquisition system is set up, details on radar system configuration, and the main preprocessing steps.

7.3.1. General Overview of the Proposed Framework

The proposed framework is depicted in Figure 7.2. For a Meta-L solution, a dataset as small and diverse as possible is generated. Specific information about the breath dataset for Meta-L is given in Section 7.3.7. The recordings from the respiration belt, which serves as a reference sensor, and the FMCW radar are collected synchronously over 30 s sessions using the recording setup described in Section 7.3.3. The preprocessing, as depicted in Section 7.3.4, aims to unwrap the phase information, reconstructing the displacement generated by user breathing. During dataset generation, the radar-based respiration signal Fc is estimated by calculating the maximum correlation between the radar phase and belt reference downstream of a double biquad band-pass filter (Sections 7.3.5.1). The entire Meta-L stage is presented in Section 7.4. From the estimated breath signal, it is also possible to calculate the instantaneous breaths per minute (bpm) and the amount of corruption per recording session caused by user motion (Section 7.3.6).

7.3.2. Radar Board and Configuration

The chosen radar system for this application is the XENSIV™ 60 GHz *BGT60TR13* FMCW, manufactured by Infineon Technologies AG [40]. The radar board is a miniaturized and low-power frequency modulated solution

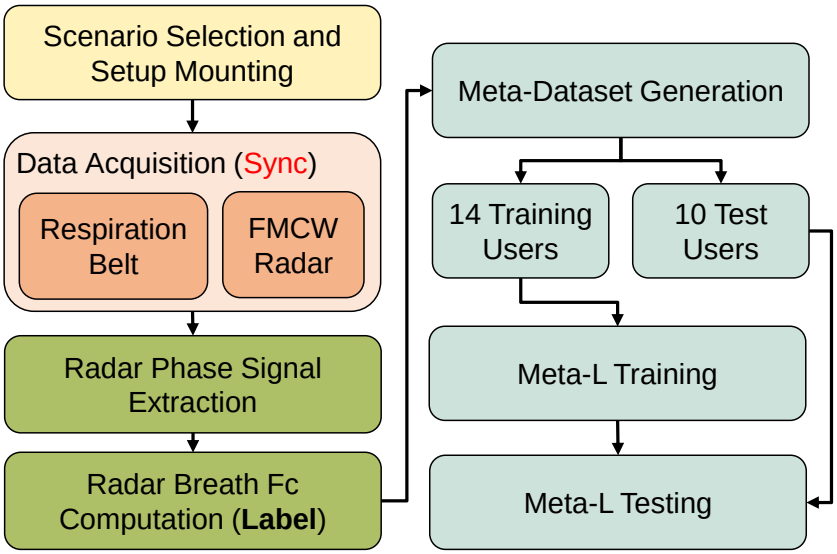


Figure 7.2: The diagram shows the main steps of the implementation. For a chosen scenario (room and user), several data sessions with synchronized radar and a reference respiration belt are collected. For multi-output ANN, the labels consist of belt reference signals and the central breath frequencies, estimated from the pure belt reference. The data from fourteen users are then used to train an ANN episodically using Meta-L, while the data from the remaining ten users are solely used for testing.

with a center frequency f_0 of 60 GHz and a bandwidth of approximately 6 GHz, which allows for a high range resolution of approximately 2 cm. In sensing applications within 5 m, the power consumption is reduced to only 5 mW thanks to an operation-optimized duty cycle. Further, by exploiting the micro-Doppler effect through phase analysis, it is possible to capture periodic displacements over time, such as vital signs, well below the 2-cm range limit [21]. The *BGT60TR13C* has three Rx channels and one Tx channel, all embedded in the package. Additionally, to enable accurate estimation of targetsázimuth and elevation angles of arrival (AoAs) in the FoV, the Rx antennas are positioned orthogonally to each other. With an f_0 of 60 GHz and a single Tx channel, such a board provides a less expensive and lower-frequency solution than many cutting-edge non-contact high-frequency vital signs systems. The evaluation board with the sensor board mounted on top is shown in Figure 7.3.

The *BGT60TR13C* generates chirps, which are a series of linearly frequency-modulated signals with a bandwidth of B_w centered on f_0. Each chirp lasts t_c and is made up of a predetermined number of n_s samples. In use, the data gathered from the Rx channels are mixed with a Tx reference and digitized with 12-bit resolution. The generated output signal is referred to as interme-

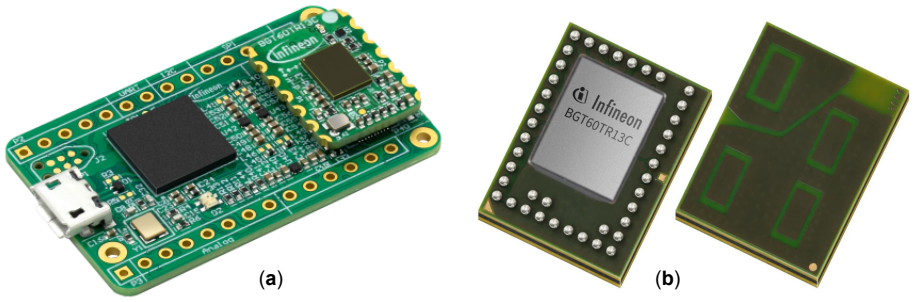


Figure 7.3: The *BGT60TR13* radar system (**a**) delivers filtered, mixed, and digitized information from each Rx channel. The *BGT60TR13C* radar (**b**) is mounted on top of the evaluation board.

diate frequency (IF). Radar data are frequently compressed into frames for additional preprocessing, with each frame carrying the IF for a series of N_c chirps. For an FMCW modulation, the theoretical range resolution Δr and maximum detection range R_{max} are calculated using the following formulas:

$$\Delta r = \frac{c}{2B_w} \ , \tag{7.1}$$

$$R_{max} = \frac{\Delta r}{2} n_s \ , \tag{7.2}$$

where c indicates the speed of light in the air. For the application of breath sensing at the workplace desk, a theoretical R_{max} of 50 cm would be sufficient. However, a theoretical maximum distance of about 3.75 m was selected for compatibility with other use cases and for future works. In the preprocessing, though, only the range bins where the user is detected are processed. The selected Δr instead is roughly 37.5 cm, which enables user identification from the surrounding clutter (static targets). The set values of B_w and n_s are accordingly 4 GHz and 200. For appropriate phase analysis, we also chose t_c and N_c values of 150 μs and 2, respectively. To acquire around 20 frames per second, a frame repetition time (fps) of 50 ms was chosen. Additionally, a 2 MHz ADC sampling rate F_s was used. All the values selected for the radar board configuration are outlined in Table 7.1.

7.3.3. Recording Setup

The recording setup, shown in Figure 7.4, is consistent with the chosen application, i.e., at-desk workplace monitoring at short distances (up to about 40 cm). The *BGT60TR13C* radar system is mounted on the front of the desk, and the Go Direct ® respiration belt [41] is placed at the level of the users'diaphragm. The belt is used as a reference to measure displacement in N (Newtons). We chose to use this belt as a reference since it is employed

Table 7.1: *BGT60TR13C* radar board, parameters configuration for breath sensing.

Symbol	Quantity	Value
N_{Tx}	number of transmitters	1
N_{Rx}	number of receivers	3
N_c	number of chirps	2
n_s	samples per chirp	200
f_0	center freq.	60 GHz
F_s	sampling freq. ADC	2 MHz
fps	frames per second	20 Hz
t_c	chirp time duration	150 µs
B_w	bandwidth	[58, 62] $\rightarrow$ 4 GHz

in other state-of-the-art work for benchmarking with radar solutions such as [36, 35]. As reported in these works, the belt has a force resolution of 0.1 Newton. This resolution allows displacements generated by breathing to be distinguishable in the presence of user motion. Although we consider the belt as a reference, such a sensor may also be subject to motion corruption. In the specific use case at the desk workplace, many of the movements made by users have little impact on a chest-mounted wearable sensor. A practical example may be typing on the keyboard. Other movements, such as bending the back, can also degrade the belt signal. In our work, however, we impose the constraint that the belt signal is the ground truth, unaffected by motion corruption noise. The data gathered by the respiration belt and radar system have been synchronized during the recordings. The data from the two sensors were synchronized frame by frame using a global time stamp generated at the laptop level. The gathering and synchronization have been performed with an Intel® Core i7-8700K CPU. Data collection was performed for 24 healthy users with an age range of up to about 35 years. All users agreed in advance to participate in data collection. The data were collected and stored as anonymously as possible, without tracking names or other characteristics that could be used to identify an individual. The data will not be made public. The users were told to behave as normally as possible, performing actions such as laughing, joking, and using the keyboard and mouse. Many users also chose to watch a video during data collection to avoid respiratory bias due to recording. For each user, 20 sessions of 30 s each were collected. Two desks in different offices and two distance ranges were chosen. The desks used are of the same type and height (about 76 cm). However, data were collected in two different environments to avoid the potential overfitting of ML models on a single location. A total of 10 sessions per user were collected at a distance from the radar board to the person's chest of up to 30 cm and

another 10 up to about 40 cm. With the 10 min allotted to each user, a total of 4 h of data was collected.

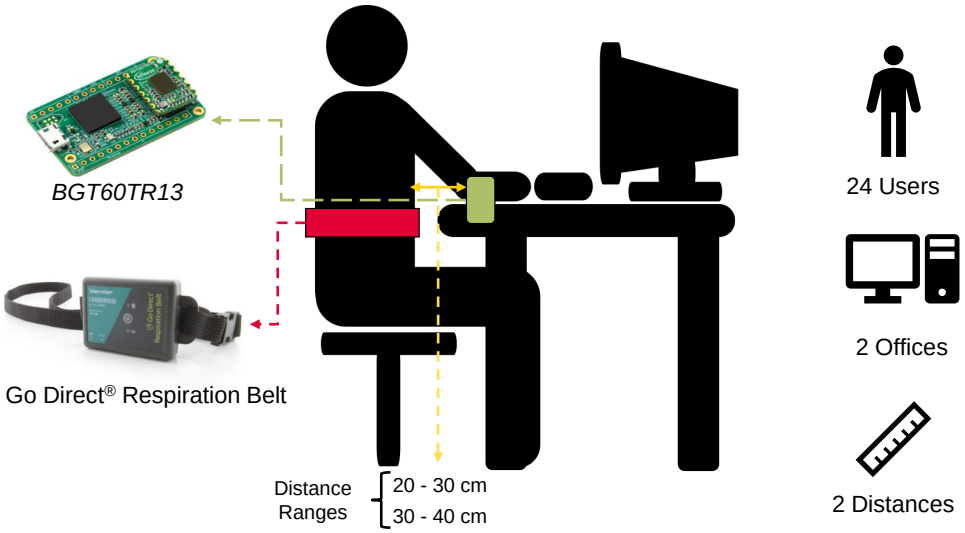


Figure 7.4: Recording Setup. A synchronized radar system and respiration belt are used to collect 10 30-second sessions per user and distance. The distance ranges used in data collection (up to 30 or 40 cm), refer to the distance between the chest and the radar board.

7.3.4. Radar Phase Signal Extraction

Thanks to the micro-Doppler effect [21], it is possible to extract the breath information from the unwrapped phase signal derived from the raw radar data. The preprocessing pipeline for the application is shown in Figure 7.5. The preprocessing can be divided into the following steps:

- Raw radar and respiration belt data are collected synchronously for a session. The chosen frame rate per session is 660 (N_m), which is 10 % higher than the theoretical frame rate of 600 (20 fps * 30 s). Longer sessions for either sensor are interpolated, whereas shorter ones are zero-padded. The belt signal is used as a reference estimation in the Meta-L training phase. Subsequent preprocessing steps involve the radar signal only.
- The IF signal is computed channel-wise, for the three Rx, for each radar frame. The information is organized in a 3D matrix, with the x-axis representing fast time (samples), the y-axis representing slow time (chirps), and the z-axis representing channels.
- The average value is subtracted from the sequence of 660 frames so that the potential direct current (DC) offset is subtracted.

- Over slow time and channels, the radar-sensed information derives from the same recorded event. Rather than using a single channel or single chirp, we use the averaged information over both axes for the next steps. Intrinsically, given the equal importance of the information in the chirps and their respective channels, the averaged information will be more robust to the noise.

- A 1D FFT is performed along fast-time to retrieve the range information.

- From the range information, it is possible to estimate the user's position frame-wise, select the set of meaningful range bins, and subtract the clutter in each (Section 7.3.5).

- The phase information is calculated for the selected bins. Frame-wise, only the bin range with the highest mean squared error (MSE) to the estimated clutter is chosen (Section 7.3.5).

- The phase beyond $(-pi, pi)$ is then unwrapped using a phase discontinuity threshold approach.

- Because users had freedom of action during the recordings, the Fc estimated from the radar phase by frequency analysis may not coincide with the central respiration Fc. For this reason, Meta-L is used to map the radar phase to the computed ideal belt Fc (Section 7.3.5.1).

- Comparison between radar-estimated breath signal and respiration belt is performed on normalized signals between zero and one, calculating MSE and estimating instantaneous bpm along the session (Section 7.3.6).

7.3.5. Range Bins Selection and Clutter Removal

Relevant radar information is only contained in a limited range of bins that reflect the user's position relative to the radar board. Let $S_R(m, s)$ with $m \in [0, N_m]$ and $s \in [0, n_s]$ be the radar signal with range information on the x-axis and slow time on the y-axis. The maximum bin range is calculated $\forall\ m$ as follows:

$$\max_{s} |S_R(m, s)| \ . \tag{7.3}$$

Around the maximum detected, 12 range bins are also processed for phase information extraction. The boundary range bins are dynamically updated via a moving average of eight frames. The dynamic adaptation avoids abrupt changes in the range under process due to instantaneous noise. Clutter is

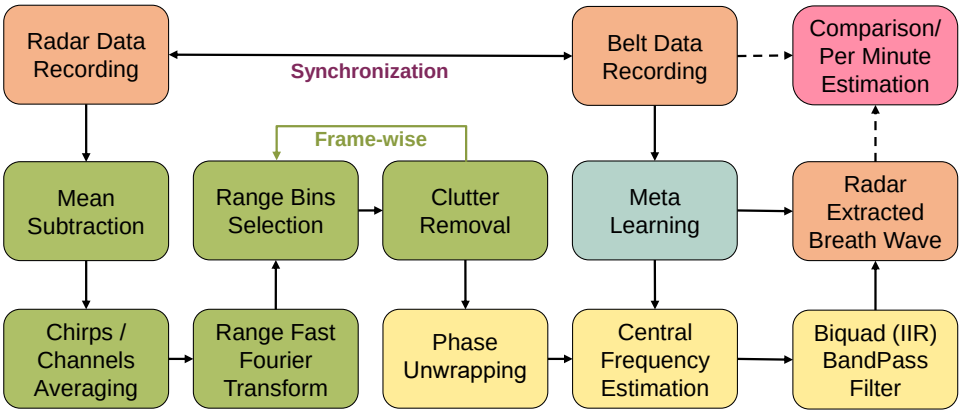


Figure 7.5: Preprocessing pipeline. First, the phase information is unwrapped from the raw radar data. The respiration signal and Fc are then estimated by Meta-L, exploiting only in the training phase the data collected with the respiration belt.

computed only for the selected bins s, frame-wise $\forall\ m \in [0, N_m]$, using the moving target indication (MTI):

$$\overline{S_R(s)}_{new} = \alpha\ S_R(m,s) + (1-\alpha)\ \overline{S_R(s)}_{old}\ ; \tag{7.4}$$

where $\alpha \in [0,1]$ is set to 0,4 and $\overline{S_R(s)}_{old}$ is the average over the preceding s bins. The value of α was chosen empirically, noting that values less than 0.2 gave too much weight to previous clutter contributions, while values greater than 0.6 depended too much on the current radar signal. Only the bin range with the highest peak-to-clutter information is extracted, which corresponds theoretically to the subject position at a given time. The MSE between $S_R(m,s)$ and the new clutter $\overline{S_R(s)}_{new}$ is calculated. The maximum MSE value corresponds to the highest peak-to-clutter. Two examples of the user range over session time are depicted in Figure 7.6.

7.3.5.1. Central Frequency Estimation and Labeling

The respiration rate can be estimated by spectral analysis in a given session, for example, by analyzing the power spectrum and taking the maximum peak in a given frequency range. Such a method is often employed for radar data recorded in idle conditions but is more of a challenge in the presence of user motion. The respiration bandwidth and Fc also depend strongly on the physiology and characteristics of the individual. Therefore, we propose using Meta-L to estimate the Fc in a user-adaptable manner, using the central frequency extracted from the respiration belt as a reference label. The reference Fc is obtained from the belt signal power spectrum by locating the frequency corresponding to the maximum peak in the limit range [0.1, 0.5] Hz, corresponding to 6 and 30 bpm. Section 7.4 describes how the proposed method

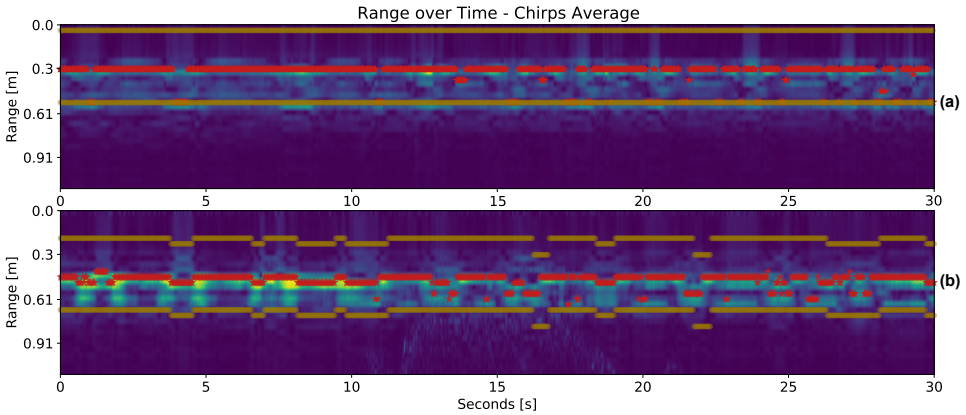


Figure 7.6: Lines in yellow indicate the defined range bin limits and, in red, the detected maximum bin per frame. Range plotting is generated after clutter removal. In (**a**), the subject did not move much during the session. In (**b**), the range limits vary according to the user's distance from the radar board.

estimates radar-based breathing signal learning from both the reference belt signal and relative Fc. The radar-based breathing signal is then computed by applying a sequence of two biquad band-pass filters to the unwrapped phase signal with the estimated Fc. The employed filter is a second-order digital recursive linear infinite impulse response (IIR) containing two poles and two zeros. A time representation of the filter can be described as follows:

$$O[n] = a_0\ I[n] + a_1\ I[n-1] + a_2\ I[n-2] - b_1\ O[n-1] - b_2\ O[n-2]\ , \tag{7.5}$$

where n is the time step; I the input vector; O the output vector, and $a_0, a_1, a_2, b_0, b_1, b_2$, the filter parameters according to the type. These latter parameters depend on the Fc selected for a given session. The formulas are provided in Appendix 7.8. Each of the two cascaded biquad filters has a quality factor Q of $\sqrt{2}$ and a sampling frequency (fs) of 20 Hz, corresponding to the fps. The characteristics of the filter are outlined in Figure 7.7.

7.3.6. Breaths per Minute Estimation and Corruption Detection

Along with the collected data session, the instantaneous bpm can be assessed via a sliding window. This information may also be useful in radar sessions that have been partially corrupted by motion and contain less respiration information. The sliding window is dynamically computed per session proportionally to the average distance between peaks. By leveraging this window, it is also possible to estimate the instantaneous motion corruption by comparing the signal with itself through autocorrelation. The corruption

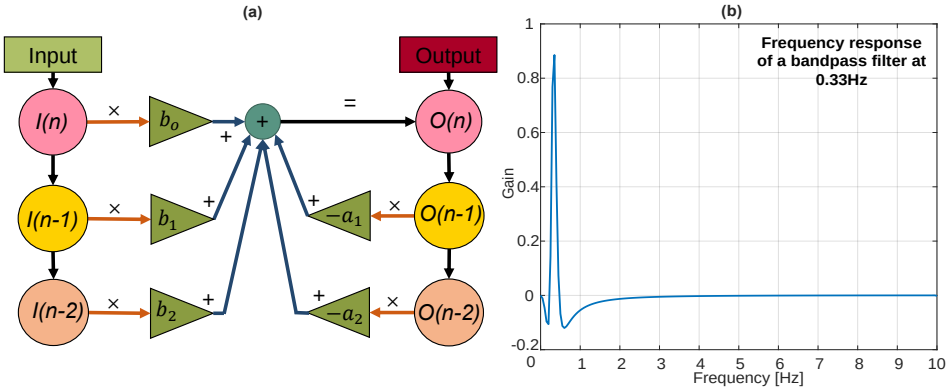


Figure 7.7: Band-pass bi-quadratic filter. The diagram (**a**) depicts the linear flow of the biquad filter, where the output $O(n)$ at time instant n is determined by the two previous input I and output O values. Instead, a gain vs. frequency plot of a biquad band-pass filter obtained for a Q of $\sqrt{2}$ and fs of 20, over an Fc of 0.33 Hz, is shown as a reference in (**b**).

information is used to weight the training samples in Meta-L and improve the predictions, as explained in Section 7.4.

In a recording session, the sliding window is defined as twice the average distance between the detected peaks of the radar phase signal after band-pass filtering. The window length is, therefore, about two whole cycles of breathing, intended as sequences of inhalation and exhalation. Because the estimated peaks for radar and belt may not match, a specific sliding window is calculated for each of the two signals. An example of the belt and radar respiration signal with computed sliding window is shown in Figure 7.8. In this instance, the respiration Fc for the radar signal is ideally extracted from the belt and has not yet been estimated by Meta-L. Local peak time shifts are visible in the plot between the radar and belt signals. These shifts are caused by two main reasons in the radar signal. First, the belt signal already contains the breathing information, whereas the radar requires multiple preprocessing steps. These steps, including the biquad filter, cause global shifts in the extracted information. In addition, the respiration belt is connected to the individual during recordings, while the radar is connected to the desk. As a result, millimeter-scale user displacements along the session can contribute to local shifts in the radar respiration signal peaks with respect to the belt. Discrepancies in amplitude, on the other hand, can be caused by ambient noise and the extracted phase, which is very sensitive to small displacements. Corruption is also visible in the radar signal at the beginning of the session. As it is not present in the belt signal, it was most likely caused by arm movements, which are mostly undetectable by the wearable sensor on the chest.

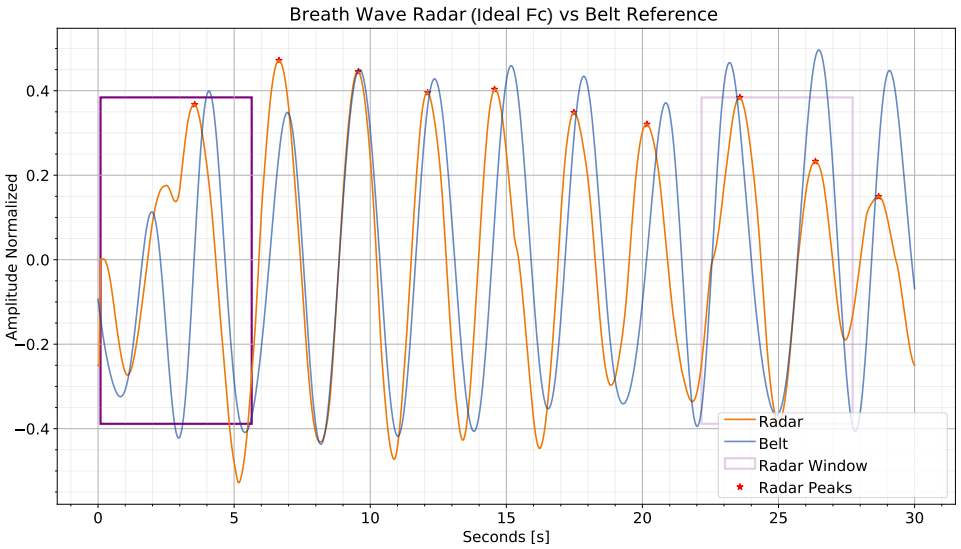


Figure 7.8: Example of sliding window generation for instant bpm estimation on a recorded session The radar signal has been filtered using the ideal belt, Fc. The radar, as opposed to the belt, is not connected to the user during recordings, but to the desk. This results in the local shift of signal breathing peaks due to the millimeter movements of the user. The window (in purple in the plot) is shown paler on the two peaks closest to the calculated peaks'mean distance. It is also possible to notice some slight corruption at the beginning of the session due to user motion.

The instantaneous bpm value is estimated as the number of peaks within the sliding window throughout the session. Because two different sliding windows are calculated for radar and belt, the length of the x-axis bpm estimate (number of samples minus the length of the sliding window) may not match in all the sessions. Autocorrelation along the session is used to estimate corruption, with a window about one and a half times as long as a breathing peak (three-quarters of a bpm sliding window). The correlation of the signal with itself gives a measure of how similar and periodic it is over time. A flag variable, by default set to zero, is set to one when the maximum autocorrelation goes below a threshold. This threshold is adjusted dynamically for the length of the sliding window. Empirically, this value is, for normalized sessions between 0 and 1, set to 0.001 times the sliding window length. On user-collected test examples, a magnitude less than 0.001 or greater seems to lead to over- or underestimation of corruption, respectively. The instantaneous bpm and corruption flag are plotted in Figure 7.9 for the same session as in Figure 7.8.

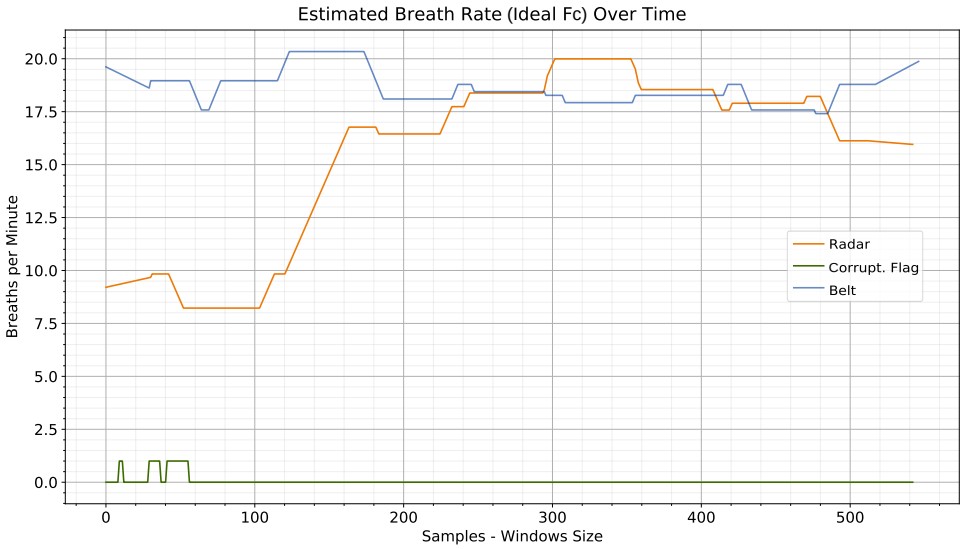


Figure 7.9: Comparison of instantaneous bpm between respiration belt and radar (with ideal Fc) for a recording session. The x-axis corresponds to the difference between the number of frames in the session and the sliding window length. The radar signal corruption flag variable is plotted in green. At the beginning of the session, the radar signal is motion-corrupted (as shown in Figure 7.8) and thus does not lead to a reliable bpm. On the other hand, for the workplace use case, the reference belt signal is more robust to motion. In this case, the motion performed was the movement of the hands toward the desk.

7.3.7. Breath Meta-Dataset

The *Breath Meta-Dataset* for training and testing the Meta-L algorithm contains data collected from 24 different healthy users up to 35 years old. Specific information on setup and data collection is provided in Section 7.3.3. For each session collected, the unwrapped radar phase signal (Meta-L input), the respiration belt signal, and the corresponding ideal respiration Fc (Meta-L outputs) are saved. All signals are interpolated or sampled to have a length of 660 samples for the 30-second recording. Both radar phase and belt signals are normalized between zero and one and translated in the interval by their average. Fourteen users were randomly selected for episodic Meta-L training, whereas the other ten were used for testing. The breath signal is highly dependent on an individual's characteristics and the presence of motion. A subject-wise two-component t-distributed stochastic neighbor embedding (t-SNE [42]) of the radar phase signal for all data collected is shown in Figure 7.10. All radar signals for t-SNE representation were filtered with a series of 2 band-pass filters with respiration frequency Fc fixed at 0.33 Hz and Q at 0.8. As can be seen from the figure, under t-SNE assumptions, two

components do not seem to be sufficient to show user-specific characteristics. This emphasizes how complex the interpretation of radar data is to extract features in an unsupervised manner for such an application.

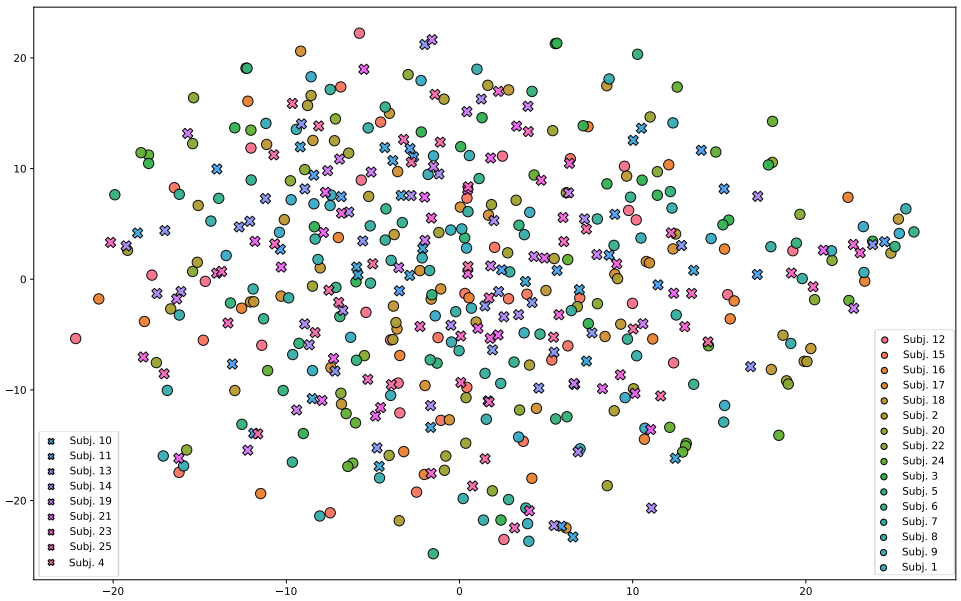


Figure 7.10: Two-component t-SNE representation of the *Breath Meta-Dataset* radar data. The circles represent the training users, while the crosses represent the testing users for the Meta-L. No user-specific feature clusters are visible under the t-SNE assumptions. The t-SNE was obtained with a perplexity of 20 and 7000 iterations [42].

7.4. Proposed Method

In this section, we describe the algorithm and topology we chose to generate the user-adaptable Meta-L model on the *Breath Meta-Dataset*. We propose an episodic learning approach by exploiting Meta-L and a C-VAE for regularized feature extraction. Once trained for generalization, leveraging a few examples gathered via the reference respiration belt, the model enables the fast adaptation of the non-contact radar sensing solution to a new user. To partially overcome the problem of motion corruption in sessions, we also present an optimized loss function (Section 7.4.3) that makes use of the corruption estimation method presented in Section 7.3.6.

7.4.1. Episodic Breath Signal Estimation

For the episodic breath signal estimation approach, we use the optimization-based MAML second-order algorithm (MAML 2^{nd}) [30]. Let R be the set

of training episodes. A task $\mathcal{T}_r$ is sampled for each $r \in R$, corresponding to a single training user. During the episode, a model learns to map the unwrapped radar phase x to the reference data of the respiration belt x_{belt} using k shots of support. The model learns by minimizing the binary cross-entropy (BCE), as follows:

$$BCE(x, x_{belt}) = -x\ log(x_{belt}) - (1 - x)\ log(1 - x_{belt}), \tag{7.6}$$

with variables x and x_{belt} in $[0, 1]$.

Radar features are encoded in a normally distributed variable $z \sim \mathcal{N}(\mu_x, \sigma_x)$ using a C-VAE topology. The μ_x and σ_x variables represent the mean and standard deviation for a given latent space dimension and input x, respectively. In addition, the Kullback–Leibler (KL) divergence [43] is used to ensure that z is close to a reference $\mathcal{N}(0, 1)$ distribution. Although KL divergence allows regularization of latent space approaching a standard multivariate normal distribution, C-VAE could learn to extract unnecessary information from the unwrapped radar phase. Such information includes, for example, displacements caused by user motion during sessions or noise. To overcome this, the latent space generation is constrained by breathing information. This is achieved by minimizing the MSE between ideal Fc extracted from the belt (y) and $\hat{y}$ predicted via a single-neuron dense layer with the linear activation function.

Adding up the three components, the loss function L is defined as follows:

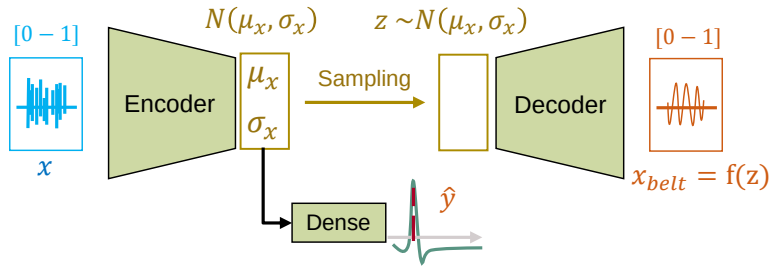
$$L(x, x_{belt}, y, \hat{y}) = BCE(x, x_{belt}) + KL[\mathcal{N}(\mu_x, \sigma_x), \mathcal{N}(0, 1)] + K\ ||y - \hat{y}||^2, \tag{7.7}$$

where K equal to 1000 is an equalization coefficient aimed at adjusting the magnitude of the MSE. The same loss function is also used in the outer step of Meta-L, on a query sample, given for the same task $\mathcal{T}_r$. The loss function terms are represented in Figure 7.11.

For a fixed training strategy, model generalization is assessed based on the ability to perform better on new tasks as episodes progress. This is performed by evaluating the model after each outer step on two evaluation tasks $\mathcal{T}_r$ and one $\mathcal{T}_v$ sampled by the training and test users, respectively. Box plots are constructed based on the loss values obtained for sequences of episodes. As the episodes progress, the mean loss should decrease, and the box plotsínterquartile range (IQR) and whiskers should also get smaller. This represents the ideal training behavior in Meta-L. Such factors, when also observed on the tasks $\mathcal{T}_v$, highlight how generalization occurs even on test users, never observed in the training phase.

7.4.2. Proposed C-VAE-Based Topology

The chosen C-VAE topology takes the unwrapped radar phase x as an input and returns two outputs. Decoder-side, the network attempts to re-



$$L = BCE(x - x_{belt}) + KL[N(\mu_x, \sigma_x), N(0,1)] + K \|y - \hat{y}\|^2$$

Signal Reconstruction | Regularization | Fc Regression

Figure 7.11: Graphical representation of single-episode learning with C-VAE. The unwrapped radar phase is mapped to the respiration belt signal using the signal reconstruction term. The regularization term makes the latent space closer to a standard multivariate normal distribution. Fc regression allows the parameterization to depend on the respiration signal.

construct the x_{belt} reference signal from the variable z. The ideal Fc, also extracted from the belt reference, is regressed from the latent space, with a single-neuron dense layer and linear activation function. The encoder contains two sequences of convolutional blocks that extract features from x. The decoder, on the other hand, tries to reconstruct x_{belt} starting from z with two deconvolution blocks and up-sampling. The C-VAE topology is shown in Figure 7.12 with an indication of layers and their parameters. The strategy of mapping the unwrapped radar phase to the belt signal attempts to counteract the problems of amplitude discrepancy and local peak time shift described in Section 7.3.6. The latent space generated during training, in fact, depends on the belt signal, which is gathered directly from the sensor attached to the individual's lower chest.

The dimension of the latent space can considerably impact the model's performance. In our experiments, we chose a dimension of 32 as the trade-off between performance and topology size. In total, the chosen topology has 739,074 parameters, all of which are trainable. Some examples of generated latent space in relation to different inputs are shown in Figure 7.13.

7.4.3. Corruption-Weighted Loss and Breathing Estimation Formulation

Even though C-VAE is set up to get information about breathing by predicting the Fc, there is still a problem when the user is moving. Radar data sessions may, in fact, be highly corrupted by motion noise and not contain the necessary respiration information. In such cases, mapping the unwrapped radar phase to the belt signal is not an effective choice. With the amount of

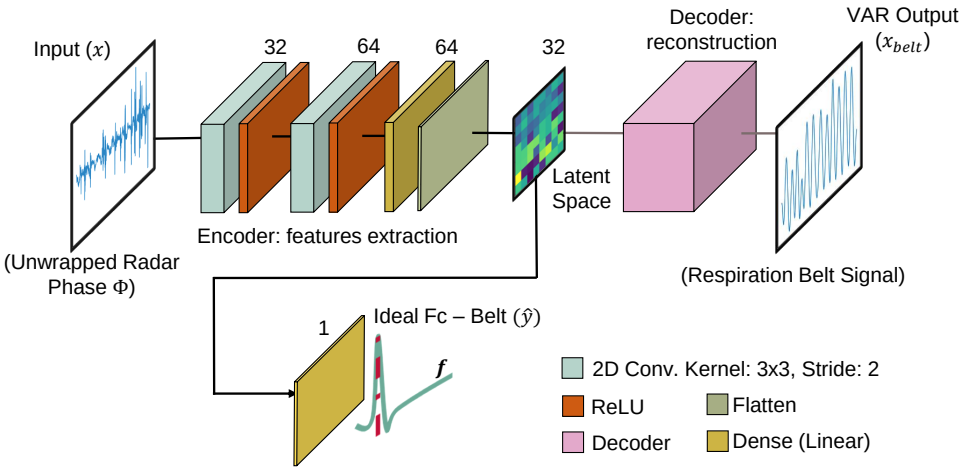


Figure 7.12: Chosen C-VAE topology. The latent space representation is constrained by both the reconstruction of x with respect to the x_{belt} reference and the ideal Fc of breathing y. The decoder layers are an up-sampled mirror version of the encoder layers.

predicted corruption motion and taking advantage of the method described in Section 7.3.6, a higher priority can be given to estimating the Fc than to reconstructing the ideal signal. The corruption rate for each session can be estimated by summing frame-wise $\forall\ m \in [0, N_m]$ the corruption flag variable $c(m)$. The greater the motion corruption, the greater the contribution of Fc in the Loss Function L must be over the signal reconstruction term.

The L can then be adjusted as follows (L^*):

$$L^* = \tau\ BCE(x, x_{belt}) + KL[\mathcal{N}(\mu_x, \sigma_x), \mathcal{N}(0,1)] + \gamma\ K\ ||y - \hat{y}||^2, \quad (7.8)$$

where $\tau = \frac{1}{\sum_m^{N_m} c(m)}$, and $\gamma = 1 - \tau$.

The τ values are obtained during Meta-L training and are normalized per epoch to the training batch size. Consistently, the reconstructed breathing signal via the C-VAE topology can be corrected using the two estimated outputs $\hat{x}_{belt}$ and $\hat{y}$ and the predicted corruption level for the single session. An adjusted estimate $\hat{x}^*$ of the radar-based breathing signal can be given by the following formula:

$$\hat{x}^* = \frac{\tau\ \hat{x}_{belt} + \gamma\ \epsilon\ biquad(x, \hat{y})}{\tau + \gamma\ \epsilon} \quad (7.9)$$

where $biquad(x, \hat{y})$ represents the filtered version of x, with the estimated $\hat{y}$ as Fc
(Section 7.3.5.1) and ϵ set to two, makes the contribution of $\hat{y}$ even more dominant in the presence of motion corruption.

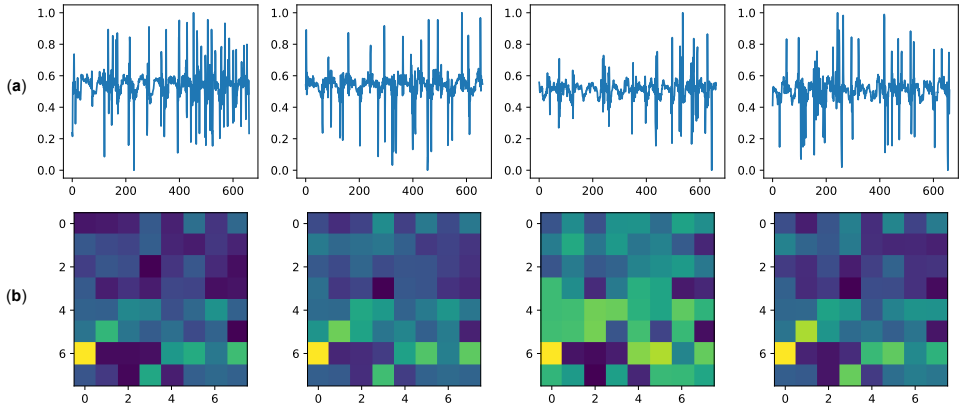


Figure 7.13: Examples of latent space generation. Examples of radar phase input (**a**) and generated latent spaces (**b**), size 32, are shown. The latent spaces are obtained after the model generalization training. Each 8 x 8 representation consists of the mean values μ and the standard deviations σ. Starting from the top of the representations toward the right, the first 32 pixels represent μ values, while the last 32 are those of σ.

7.4.4. Information about Experiments

All experiments have been conducted by minimizing the loss function L^*, training the C-VAE model with latent dimension 32. As described in Sections 7.4.1 and 7.4.3, the loss function includes a contribution to the FC as well as a contribution to the reconstruction of the respiration signal. The latter imposes normalized signals in the range $[0, 1]$. The loss consequently has no unit of measurement but can be understood as an absolute value to be minimized with respect to zero. The optimizer chosen is Adam, with β_1 and β_2 equal to 0 and 0.5, respectively. The training is performed on 3000 episodes at 4 epochs per episode. For 1–shot experiments, the batch size is one, and for 5– and 10–shot experiments, it is 5. The inner learning rate is set to $18e-4$ for the 1–shot experiments and $8e-4$ for the 5– and 10–shot experiments to avoid episodic overfitting. The chosen outer learning rate is $17e-4$. Each experiment is performed three times. The performance of the C-VAE model is evaluated in terms of mean L^* as episodes progress, adaptation time on new users, and single inference time on a new sample. For the loss evaluation, we also present a confidence value. This value represents the 95 % confidence that the true mean is included in the distribution. In general, the lower this value, the more stable and precise a given type of experiment is. The adaptation time per user corresponds to the time required by the Meta-L model to complete an entire training episode via a simple first-order gradient descent. Optimization is performed for a specific number of epochs and batches by minimizing the loss of L^* over k training shots. For the adaptation time estimation, we chose four epochs of training with first-order

gradient optimization. The other hyperparameters remain unchanged from the Meta-L training procedure. Single-sample inference time is calculated at the end of each adaptation training, on a random sample of tests for a given user. This value is consequently independent of the number of training shots selected in the adaptation. At the end of each adaptation test episode, the model parameters are restored to the values learned during the Meta-L training.

7.5. Results and Discussion

This section presents the results of Meta-L experiments on the *Breath Meta-Dataset*. The experiments were carried out with the optimization-based algorithm (MAML 2^{nd}), for 1–, 5– and 10–shots (Section 7.5.1). Without, to the best of our knowledge, any state-of-the-art Meta-L solutions for breath sensing, we compare our method to other state-of-the-art Meta-L algorithms, taking advantage of our proposed C-VAE topology (Section 7.5.3). We then show in an ablative study the benefits of using motion corruption estimation in the loss function and the model performance with various latent dimensions (Section 7.5.2).

All experiments were performed on Intel® Core i7-8700K CPU, and DIMM 16 GB DDR4-3000 module of RAM.

7.5.1. Results on MAML Second Order

All results presented represent the average of the results achieved in the various repetitions. The model performance was evaluated every 300 episodes, creating a box plot on the collected loss values in the evaluation loop over 10 test examples per class (Section 7.4.1). The episodic learning trend on a 1–shot experiment is shown in Figure 7.14. As the episodes progress, L^* decreases, as well as the IQR and whiskers. While learning from only one example per user, the model can generalize better thanks to prior acquired experience. This behavior is observable not only for $\mathcal{T}_r$ tasks but also for $\mathcal{T}_v$ test tasks, which are unobserved in episodic learning.

Box plots are an effective way to assess the progress of episodic learning but do not reveal the underlying distribution of the L^* variable. The histograms built on L^* for a given interval of episodes can help estimate the distribution and assess the generalization. The histograms corresponding to the box plots generated for the first and last 300 episodes of a 1–shot experiment are depicted in Figure 7.15. The bottom plots represent the density histogram of the L^*, while the Gaussian approximation of the box plots with their respective quartiles is shown in the middle plots. The histograms do not undergo a Gaussian distribution. At the beginning of episodic learning, the distribution is usually multimodal because of the different learning comple-

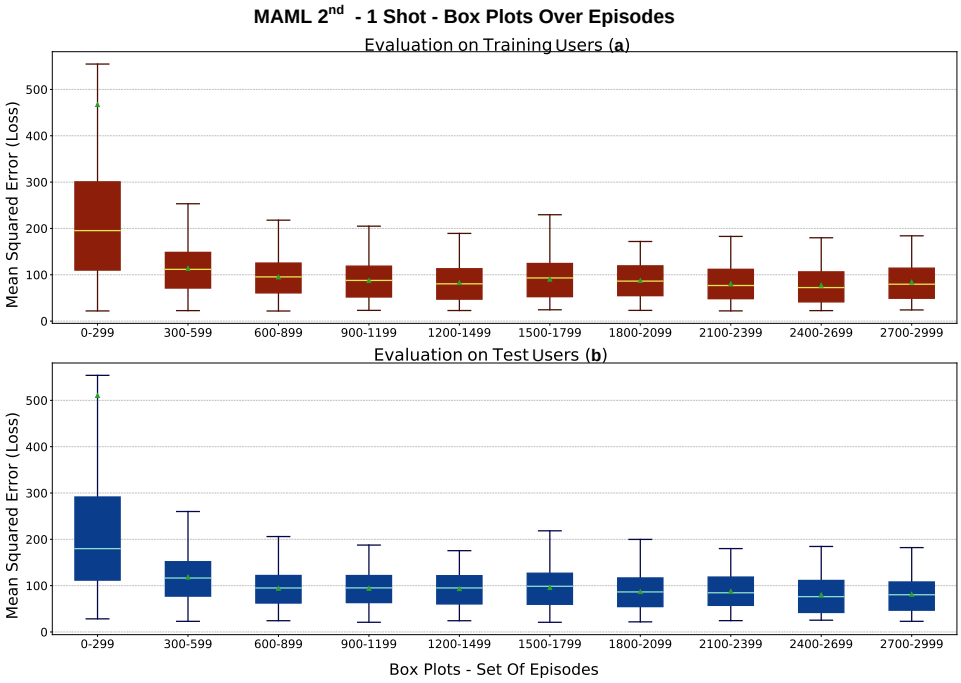


Figure 7.14: MAML 2^{nd} 1–shot experiment, Box Plots. Learning trends of Meta-L, box plots versus episodes (evaluation loop) for the *Breath Meta-Dataset*. The box in (**a**) depicts the trend for users in the training set ($\mathcal{T}_r$ tasks). In (**b**), the trend for the users of the test set ($\mathcal{T}_v$ tasks) is shown. The box's mid-line represents the median value, while the little green triangle represents the mean.

xity between tasks. In the last learning step, the histograms typically feature a positive skewness toward the zero of the L^*. This behavior occurs thanks to the generalization of information acquired episodically.

The values of L^* obtained for MAML 2^{nd} experiments for 1–, 5–, and 10–shots on test users are presented in Table 7.2.

Table 7.2: MAML 2^{nd} experiments, average L^* over the last 300 episodes of test tasks $\mathcal{T}_v$ evaluation, averaged over 3 repetitions with 95 % confidence intervals.

Loss / N–Shots	**1–Shot**	**5–Shots**	**10–Shots**
L^*	84.11 ± 6	83.92 ± 1	83.39 ± 1

As can be seen from the table, as the number of shots increases, there are no significant reductions in the mean loss for new users. Experiments with 5– and 10–shots, however, show a lower 95 % confidence value, and thus higher

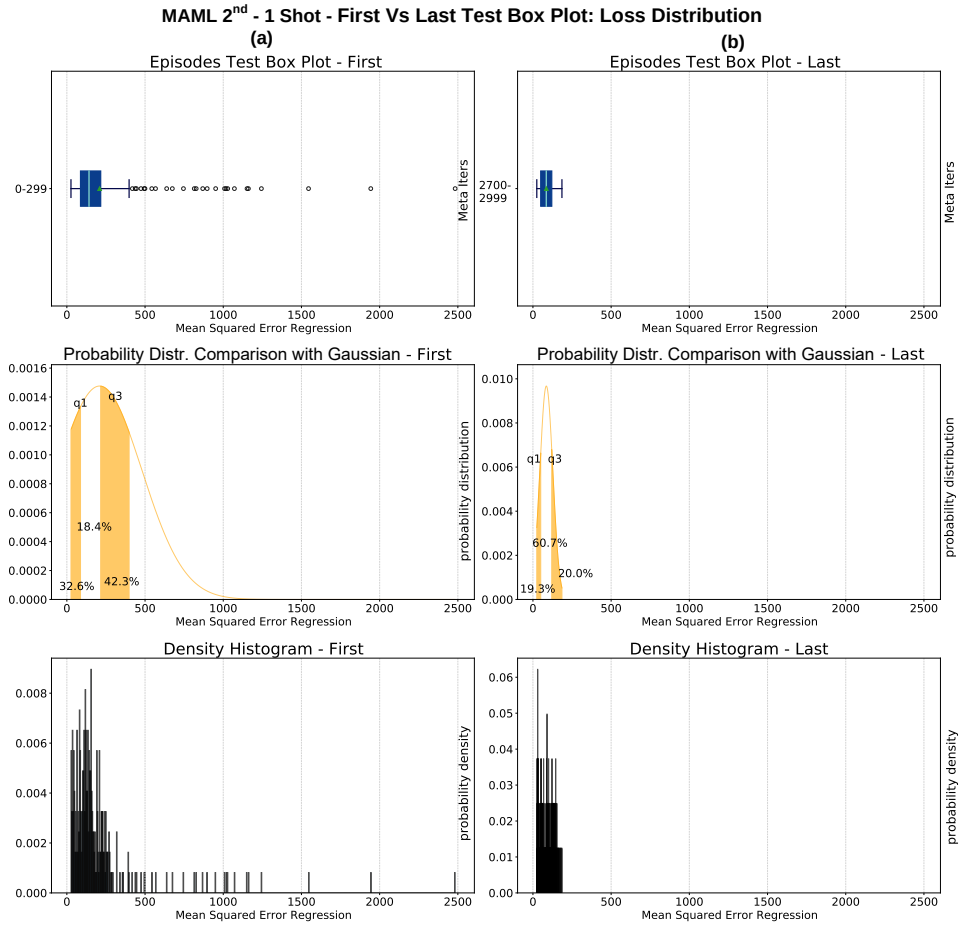


Figure 7.15: MAML 2^{nd} 1–shot experiment histograms for the first (**a**) and last (**b**) set of 300 episodes. The box plots in the topmost plots also contain outliers as small circles outside the whiskers. The mid-plots show an approximation to the Gaussian distribution. The lower plots show the true histograms, which do not underlie a Gaussian distribution. The q1 and q3 represent the first and third quartiles, respectively.

precision. This attests that in some cases, a single training example with different characteristics from others gathered may not be sufficient to generalize on the test. The generalization strategy allows the model to extract a substantial amount of breath signal information, independent of the number of shots, as illustrated in Figure 7.14. Many user data sessions are corrupted by motion and limit the increase in performance of the C-VAE model as the number of training examples increases. However, by relating the loss L^* to the average breathing rate per 30 s session, a specific learning behavior can be observed. The box plots generated according to the respiration rate for all test users are shown in Figure 7.16. The respiration rate between 7

and 9 represents a standard human breathing rate. As can be seen in the figure, the base of the box plots is not uniform over the RR range. In fact, most of the collected examples have a number of respiration peaks that are close to or equivalent to the standard value. Between 1 and 4 and 12 and 14, there are only 4 and 5 examples, respectively. On the other hand, for a respiration rate between 7 and 9, there are about 30 instances per class. This motivates the choice of the box plot construction. By fitting the model to each test user separately after episodic learning and using the remaining sessions as tests for each fit, it is possible to obtain the model behavior as a function of the RR. With the 1–shot fit, the model is more accurate at reconstructing breath signals between 7 and 9 peaks per 30 s. For lower or higher rates, the model performs less well in reconstruction, resulting in an increased loss. This behavior can be due to two main reasons. The first is that motion corruption may not have allowed the identification of the correct breathing peaks in the test sessions. Motion corruption results in the erroneous identification of breathing patterns and subsequent missed learning. The second is that the model did not have enough reference examples for low and high rates during generalization learning. Because of this, a few examples of training for a new user may not be sufficient for adaptation. For the 5– and 10–shots fit, the model seems to be able to tackle low or high RR situations better, but it performs slightly less well than the 1–shot model for standard RRs. This may be mainly caused by motion corruption in many sessions for all users. Although the L^* is defined to address such a problem, additional training examples may not contain enough information to overcome motion corruption.

Figures 7.17 and 7.18 show examples of prediction after test user adaptation of MAML 2^{nd} 1–shot. In both figures, the top plots represent the breath signal prediction while the bottom plots represent the instantaneous bpm estimation. The prediction of breath signals is obtained using the $\hat{x}^*$ formula (Equation (7.9)). Figure 7.17 depicts two examples of correct breath signal prediction throughout the session. In the example (a), there is almost no corruption due to user motion for most of the session. In accordance with Figure 7.16 for 1–shot, this example falls within the range of 12—14 beats for 30 s. Even so, the algorithm leads to a quite accurate bpm estimation, with an average gap between the belt reference and the estimated radar, of three beats. The presence of many detected peaks often corresponds to much motion corruption for radar. In this case, however, many peaks are also visible in the belt reference. The example (b), which is part of the samples in the range 7–9 peaks, is characterized by more radar signal corruption with respect to (a). Nevertheless, the robust formulation of L^* allows the extraction of peak position and bpm despite the presence of motion corruption. However, for both plots, there are discrepancies in time shift and peak amplitude between radar and belt. As described in Sections 7.3.6 and 7.4.2, the

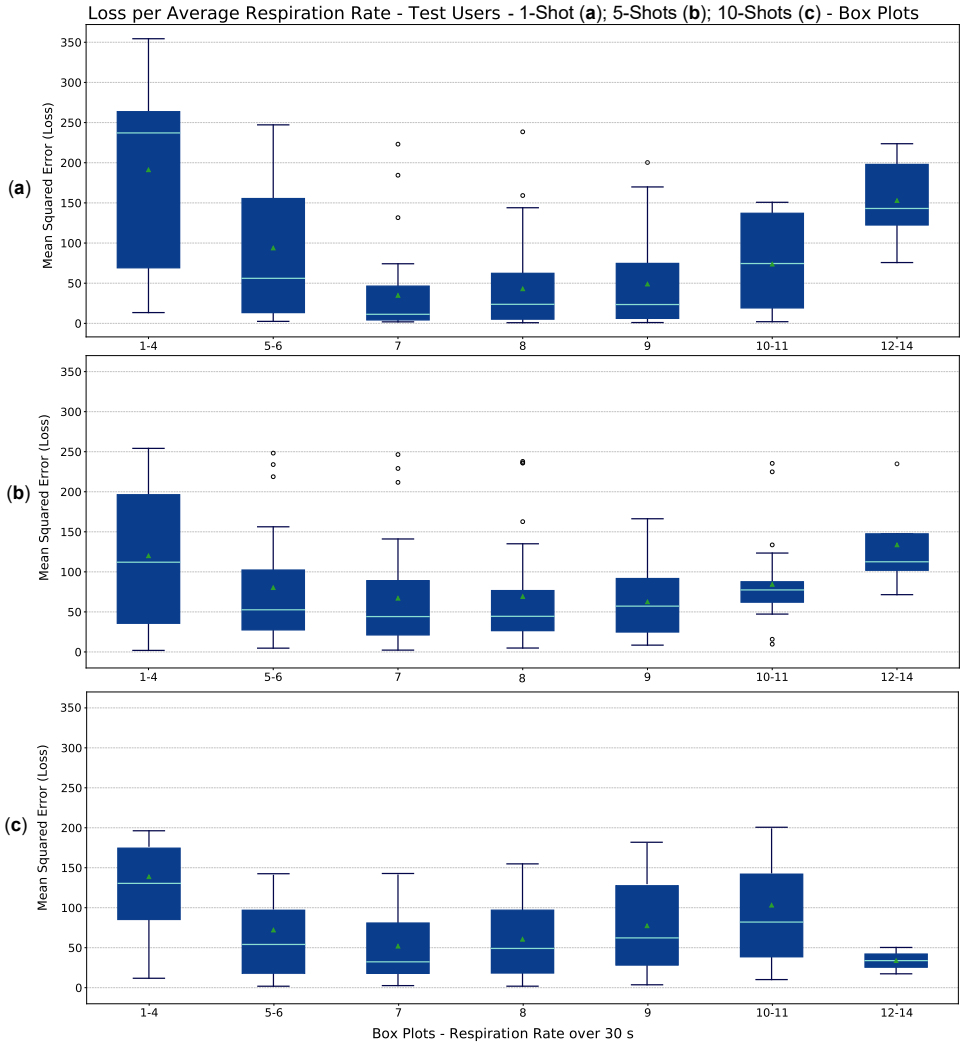


Figure 7.16: Loss (L^*) as a function of the number of detected breathing spikes over the 30 s sessions for the 10 test users. The base of the box plots with non-uniform ranges was chosen so as to have at least 4 examples for the least common classes (1–4 and 12–14). The upper plot is obtained by fitting the 1–shot Meta-L model (**a**) to new users, while the middle and lower plots are obtained by 5– (**b**) and 10– (**c**) shots adaptation, respectively. For the first two plots, the circles that lie outside the box plots whiskers represent the outliers. Plot (**c**) shows no visible outliers.

proposed C-VAE topology tries to mediate these challenges but still cannot perfectly reconstruct the belt reference signal. In general, incorrect detection of radar peaks can result in erroneous local predictions of the bpm. In both plots, the corruption flag correctly predicts the motion in correspondence to

the false peaks detected. Figure 7.18 instead shows two edge examples with a relatively low (a) and high (b) belt reference number of peaks per session. In both cases, the model leads to quite different results from the reference ones. Although the bpm estimate does not deviate much from the reference, peaks are detected at incorrect times. For (a), the combination of just a few breathing peaks and motion corruption makes correct prediction challenging. One way to potentially solve this issue would be to collect a lot of edge data and train the model episodically to generalize better in such scenarios. In the example (b), the user breathed much more frequently than in the other sessions, including the training one. This leads to the model's inability, given the prior acquired knowledge, to generalize to the user scenarios with only one training shot. The time shift between radar peaks is also not seen as corruption by the specific flag, as it is probably not caused by motion.

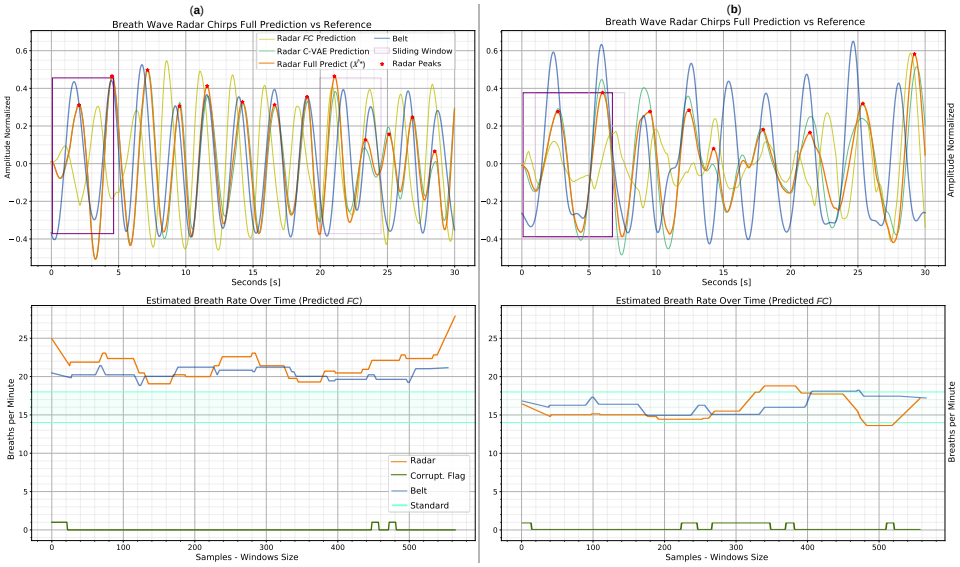


Figure 7.17: Standard prediction examples obtained post 1–shot test user-adaptation with MAML 2^{nd}. The top plots show the prediction $\hat{x}^*$ versus the respiration belt reference, while the bottom plots display the estimated bpm and corruption flag. Legends, which also apply to the plots on the right, are placed in the plots on the left. An example of optimal prediction with radar information characterized by little motion corruption is shown in (**a**). The respiration signal is recovered even in the presence of some corruption, as in (**b**), thanks to the L^* formulation.

Table 7.3 lists the adaptation time for a new user, varying the number of shots in milliseconds using L^*. The procedure of estimating the adaptation time to new users is explained in Section 7.4.4. Given four epochs of learning per single user, the algorithm requires a gradually increasing adaptation time as the number of shots increases. Indeed, compared to a 1–shot, the

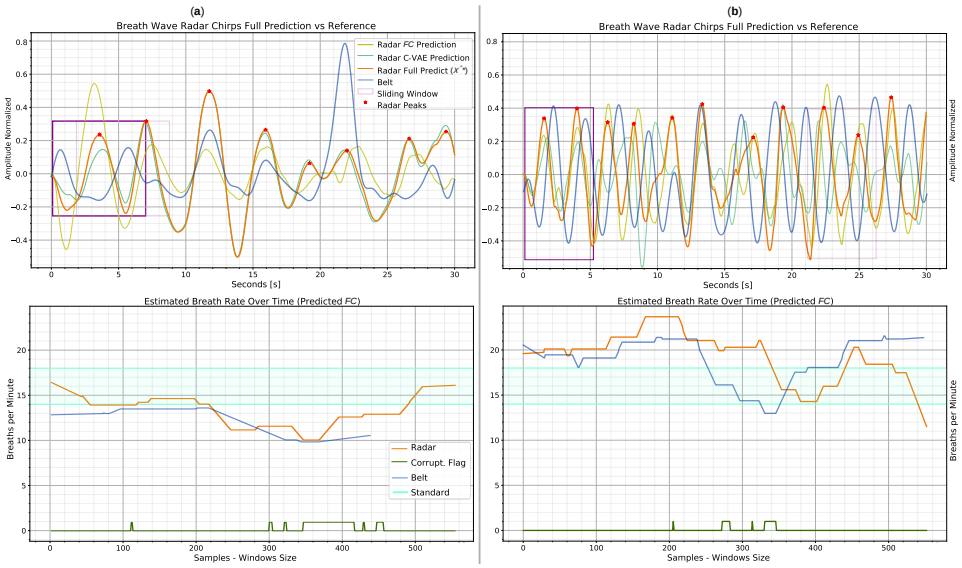


Figure 7.18: Edge prediction examples obtained post 1–shot test user-adaptation with MAML 2^{nd}. The top plots show the prediction $\hat{x}^*$ versus the respiration belt reference, while the bottom plots display the estimated bpm and corruption flag. Legends, which also apply to the plots on the right, are placed in the plots on the left. In (a), there are six visible peaks in the belt signal (blue), while in (b) there are thirteen peaks. In these examples, the algorithm performs less well than in standard cases. This is mainly due to the lack of edge data as prior knowledge during episodic learning. In the bpm estimation in the example (a), a shorter estimate can be seen for the belt than for radar. This is due to the computation of two distinct windows between radar and belt, as explained in Section 7.3.6.

adaptation time is roughly 3 and 7 times longer for 5– and 10–shots. Since the mean L^* does not decrease much for 5– or 10–shots (less than 1 %), the 1–shot model can be considered the best trade-off. On the other hand, for the 1–shot experiments, there is a bigger variation in the confidence value (up to 5 %). For this reason, we decided to show the 1–shot outcomes in the single-experiment analysis.

The single inference time for MAML 2^{nd} experiments , as discussed in Section 7.4.4, is independent by the number of shots used for user adaptation. Accordingly, we calculated an average value over all repetitions of the 1–, 5– and 10–shots experiments, already averaged over the last 300 test evaluations of each. The computed value of single inference time for MAML 2^{nd} is 4.30 ms.

Table 7.3: MAML 2^{nd} experiments, average adaptation time over the last 300 episodes of test tasks $\mathcal{T}_v$ evaluation, averaged over 3 repetitions using L^*, in milliseconds.

Time N–Shots	**1–Shot**	**5–Shots**	**10–Shots**
Adaptation Time [ms]	797	2,614	5,877

7.5.2. Ablation Study

To show the real benefits of robustness to motion corruption, it is also important to compare it with training that does not take this information into account when figuring out the loss function. This can be conducted by comparing the results obtained with L^*, with the loss L presented in Section 7.4. The average values over three experiments for loss comparison are given in Table 7.4. As can be seen from the results, the mean L^* turns out to be less than half despite the fact that the two formulated losses are characterized by the same magnitude for the three components to minimize. Moreover, the 95 % confidence shows that the loss L does not become more accurate as the number of shots increases. Probably, without limiting the learning for corrupted training examples, as for L^*, the model also learns information that is not useful for respiration estimation.. Aside from the values, restricting feature extraction to pure breathing information improves learning and model prediction by incorporating the amount of motion corruption in loss formulation.

Table 7.4: MAML 2^{nd} experiments, average L and L^* over the last 300 episodes of test tasks $\mathcal{T}_v$ evaluation, averaged over 3 repetitions, with 95 % confidence intervals.

Loss / N–Shots	**1–Shot**	**5–Shots**	**10–Shots**
L (No Corrupt.)	226.30 ± 5	224.53 ± 5	221.97 ± 5
L^* (Corrupt.)	84.11 ± 6	83.92 ± 1	83.39 ± 1

Another important feature to analyze is how the performance of the chosen C-VAE topology varies as a function of the model size. This can be accomplished by varying the size of the latent space, which represents the size of the extracted features. MAML 2^{nd} 1–shot experiments were carried out with latent space values ranging from 16 to 128. Table 7.5 shows the values of L^* and the number of trainable parameters as a function of the latent dimension. The mean L^* reaches the minimum for a latent dimension of 32, which was also selected for all experiments in Section 7.5.1. A latent dimension of 16 seems to not be enough to extract all useful breathing features from the radar phase. A dimension of 64 brings similar mean values to

32 at the expense of twice as many parameters. Looking also at the values at 95 %, both a latent space of 64 and 128 lead to a visible degradation of precision. This means that the features extracted for many of the evaluation episodes, tend to overfit the training data and thus fail to generalize well to test users.

Table 7.5: MAML 2^{nd} 1–shot experiments, average L^* and trainable parameters with varying latent dimension. The L^* values are obtained over the last 300 episodes of test tasks $\mathcal{T}_v$ evaluation. The results are provided with 95 % confidence intervals, averaged over 3 repetitions.

Parameters / Latent Dim.	**16**	**32**	**64**	**128**
L^*	86.75 ± 5	84.11 ± 6	84.31 ± 14	85.19 ± 34
Trainable Params.	382,658	739,074	1,451,906	2,877,570

7.5.3. Results on Various Optimization-Based Algorithms

Using the same C-VAE topology, the performance of MAML 2^{nd} Order can be compared to that of other cutting-edge Meta-L algorithms.

We propose comparing MAML 1^{st} (first-order model) [30], Reptile [44], and an improved in-training stability version of MAML based on some contributions from Antoniu et al. [45]. We call MAML $^+$, the stabilized version of MAML that incorporates multi-step loss optimization (MSL), derivative-order annealing (DA), and meta-optimizer learning rate cosine annealing (CA). We trained such algorithms following the same episodic training and evaluation setup as defined in Section 7.5.1, for MAML 2^{nd}. Reptile episodic training was carried out with a batch size of 2 and an inner learning rate of $3e-5$. The outer step weight update has been conducted with a meta step size of 0.4. For MAML $^+$, a value of $17e-4$ is chosen as the initial value for the outer step learning rate before cosine annealing.

The average accuracy values for test users over three repetitions of the experiment are given in Table 7.6. For 1– and 5–shots, MAML algorithms perform better than Reptile. MAML 2^{nd} produces the lowest average value of L^* for a single shot, allowing for better reconstruction of respiration signals. For 5–shots, the MAML $^+$ algorithm guarantees the best average result. The latter, however, lacks precision, leading to a broad 95 % confidence interval in the 10–shot approach and even decreasing the learning rate in the inner step. Most likely, second-order learning, coupled with training that tends to be more selective as episodes progress, leads the model to give more weight to motion corruption features. This leads to instability when multiple training samples are employed and thus decreases performance. MAML 2^{nd} and MAML 1^{st} are tied as the algorithms with the lowest mean $L*$ for 10 shots. For all algorithms except Reptile, it can be seen that there is no marked

decrease in the mean L^* as the number of shots increases. By having more data available, this behavior could be countered by using only the least corrupted data for training.

Table 7.6: Optimization-based experiments comparison, average L^* over the last 300 episodes of test tasks $\mathcal{T}_v$ evaluation, averaged over 3 repetitions with 95 % confidence intervals.

Algorithm N–Shots	**1–Shot**	**5–Shots**	**10–Shots**
Reptile	100.02 ± 2	90.78 ± 2	86.95 ± 1
MAML 1^{st}	86.52 ± 5	83.68 ± 1	83.45 ± 1
MAML $^{+}$	85.86 ± 10.7	82.9 ± 3	88.16 ± 15
MAML 2^{nd}	84.11 ± 6	83.92 ± 1	83.39 ± 1

The adaptation procedure adopted for the other algorithms is the same as that of MAML 2^{nd}, illustrated in Section 7.4.4. As the algorithms vary, only the procedure for computing the generalization parameters in the Meta-L stage changes. The parameters in the evaluation phase are algorithm specific, but the adaptation always employs first-order gradient optimization. This means that the adaptation time is independent of the chosen algorithm, since what changes are only the values of the parameters. Thus, there is no real difference in adaptation time between the chosen algorithms with respect to the value provided in Section 7.5.1. The same is true for the single-sample inference time.

7.6. Conclusions

In this paper, we present a user-adaptable and non-contact solution for respiration signal estimation using a 60 GHz FMCW radar. This system is mainly intended for office work-desk applications in a distance range of 20 to 40 cm, characterized by little user motion. This solution, while not as accurate as user-contact estimation approaches, shows how radar can potentially be employed to non-contact monitor respiration rates. The episodic learning approach eases the system's adaptation to new users through short model adaptation sessions. The estimated respiratory rate may be used for anomaly detection related to the specific user to whom the system is tailored. Thanks to a variational autoencoder, the topology employed can extract respiration features from the radar phase signal, using as a reference for reconstruction, the signal collected with a respiration belt. Although the belt could be used on its own for respiration estimation, it would not allow non-contact estimation. Through this approach, the belt can serve only in user-specific learning to enhance radar predictions. The cost function of the model is suitably modified by constraining feature generation to the respiration in-

formation, to avoid learning motion corruption information. In addition, a direct estimation of corruption in the collected data sessions allows for improved learning and model estimation in breath signal generation. The whole system presented represents the first step toward a possible non-contact solution for estimating multiple vital parameters that is adaptable quickly and has cutting-edge performance for new users. The radar solution by sensing millimeter displacements could also be used for estimating cardiac signals or the presence of muscle tremors caused by potential diseases.

Although this solution offers several innovative advantages, it also has the disadvantage of relying, only during adaptation, on a breathing belt used as a reference. We placed the constraint so that such a reference sensor depends little on degradation caused by user motion. The generated models also do not perform particularly well for users with respiratory rates significantly higher or lower than the standard 7 to 9 beats per 30 s. This is mainly due to only a few reference examples available in meta-learning, not enough for proper generalization. Radar information is also easily corrupted by long movements in the recording sessions Further, the use of multiple training examples for user adaptation results in improvements for out-of-standard respiratory rates but can degrade performance within the standard range itself. Therefore, substantially motion-corrupted sessions should still be discarded and not used for adaptation. Sensor fusion systems and discarding corrupt sessions could improve performance under these circumstances.

Future work will focus on benchmarking the presented approach against other non-contact solutions, comparing the Meta-Learning solution with transfer learning and adapting the system to other environments, such as outdoors. An additional important aspect that will be analyzed is the variation in model performance per user and bpm as the level of motion corruption in the sessions changes. Another intriguing possibility would be to test and improve the solution on users with respiratory dysfunction in order to assess its benefits and drawbacks.

7.7. Acknowledgments & Declarations

- **Funding.** The work presented is partially supported by the ITEA3 Unleash Potentials in Simulation (UPSIM) project (N°19006) funded by the German Federal Ministry of Education and Research (BMBF), the Austrian Research Promotion Agency (FFG), the Rijksdienst voor Ondernemend Nederland (Rvo) and the Innovation Fund Denmark (IFD).

- **Informed Consent.** Informed consent was obtained from all subjects involved in the study.

- **Data Availability.** The data are not publicly available due to internal company board policy.

- **Conflicts of Interest.** The authors declare no conflicts of interest.

7.8. Appendix A. Biquad Filter Parameters Computation

A biquad band-pass filter is used for filtering the respiratory signal. A temporal representation of the filter is given in Equation (7.5). The parameters $a_0, a_1, a_2, b_0, b_1, b_2$ depend on the central frequency FC, the sampling frequency fs, and Q the quality factor. Q determines the sharpness of the filter. Let ω be the amount of degrees to advance the periodic signal per sample:

$$\omega = 2\pi Fc/fs. \tag{7.10}$$

Let η be a function of ω with respect to the Q value:

$$\eta = \frac{sin(\omega)}{(2Q)}. \tag{7.11}$$

For a biquad band-pass filter, the time parameters can be calculated with the following formulas:

$$a_0 = 1 + \eta, \tag{7.12}$$

$$a_1 = -2cos(\omega), \tag{7.13}$$

$$a_2 = 1 - \eta, \tag{7.14}$$

$$b_0 = \eta, \tag{7.15}$$

$$b_1 = 0, \tag{7.16}$$

$$b_2 = -\eta. \tag{7.17}$$

References

[1] Michaela Sidikova, Radek Martinek, Aleksandra Kawala-Sterniuk, Martina Ladrova, Rene Jaros, Lukas Danys, and Petr Simonik. Vital sign monitoring in car seats based on electrocardiography, ballistocardiography and seismocardiography: A review. *Sensors*, 20(19):5699, 2020.

[2] Takunori Shimazaki, Daisuke Anzai, Kenta Watanabe, Atsushi Nakajima, Mitsuhiro Fukuda, and Shingo Ata. Heat stroke prevention in hot specific occupational environment enhanced by supervised machine learning with personalized vital signs. *Sensors*, 22(1):395, 2022.

[3] Patrick C Loughlin, Frank Sebat, and John G Kellett. Respiratory rate: The forgotten vital sign—make it count! *Joint Commission Journal on Quality and Patient Safety*, 44(8):494–499, 2018.

[4] Idar Johan Brekke, Lars Håland Puntervoll, Peter Bank Pedersen, John Kellett, and Mikkel Brabrand. The value of vital sign trends in predicting and monitoring clinical deterioration: A systematic review. *PloS one*, 14(1):e0210875, 2019.

[5] World Health Organisation. Cardiovascular diseases (cvds), 2021. `https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)`, Last accessed on 2022-10-25.

[6] Duarte Dias and João Paulo Silva Cunha. Wearable health devices—vital sign monitoring, systems and technologies. *Sensors*, 18(8):2414, 2018.

[7] Zhihua Wang, Zhaochu Yang, and Tao Dong. A review of wearable technologies for elderly care that can accurately track indoor position, recognize physical activities and monitor vital signs in real time. *Sensors*, 17(2):341, 2017.

[8] William Taylor, Qammer H Abbasi, Kia Dashtipour, Shuja Ansari, Syed Aziz Shah, Arslan Khalid, and Muhammad Ali Imran. A review of the state of the art in non-contact sensing for covid-19. *Sensors*, 20(19):5665, 2020.

[9] Ruchika Sinhal, Kavita Singh, and Anuraj Shankar. Estimating vital signs through non-contact video-based approaches: A survey. In *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*, pages 139–141. IEEE, 2017.

[10] Mauricio Villarroel, João Jorge, Chris Pugh, and Lionel Tarassenko. Non-contact vital sign monitoring in the clinic. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 278–285. IEEE, 2017.

[11] Marc Garbey, Nanfei Sun, Arcangelo Merla, and Ioannis Pavlidis. Contact-free measurement of cardiac pulse based on the analysis of thermal imagery. *IEEE transactions on Biomedical Engineering*, 54(8):1418–1426, 2007.

[12] Toshiaki Negishi, Shigeto Abe, Takemi Matsui, He Liu, Masaki Kurosawa, Tetsuo Kirimoto, and Guanghao Sun. Contactless vital signs measurement system using rgb-thermal image sensors and its clinical screening test on patients with seasonal influenza. *Sensors*, 20(8):2171, 2020.

[13] Michele Ambrosanio, Stefano Franceschini, Giuseppe Grassini, and Fabio Baselice. A multi-channel ultrasound system for non-contact heart rate monitoring. *IEEE Sensors Journal*, 20(4):2064–2074, 2019.

[14] Mamady Kebe, Rida Gadhafi, Baker Mohammad, Mihai Sanduleanu, Hani Saleh, and Mahmoud Al-Qutayri. Human vital signs detection methods and potential using radars: A review. *Sensors*, 20(5):1454, 2020.

[15] Anuradha Singh, Saeed Ur Rehman, Sira Yongchareon, and Peter Han Joo Chong. Multi-resident non-contact vital sign monitoring using radar: A review. *IEEE Sensors Journal*, 21(4):4061–4084, 2020.

[16] Jian Liu, Hongbo Liu, Yingying Chen, Yan Wang, and Chen Wang. Wireless sensing for human activity: A survey. *IEEE Communications Surveys & Tutorials*, 22(3):1629–1645, 2019.

[17] Takamochi Kanda, Takashi Sato, Hiromitsu Awano, Sota Kondo, and Koji Yamamoto. Respiratory rate estimation based on wifi frame capture. In *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, pages 881–884. IEEE, 2022.

[18] Fengyu Wang, Feng Zhang, Chenshu Wu, Beibei Wang, and KJ Ray Liu. Vimo: Multiperson vital sign monitoring using commodity millimeter-wave radio. *IEEE Internet of Things Journal*, 8(3):1294–1307, 2020.

[19] Graham M Brooker et al. Understanding millimetre wave fmcw radars. In *1st international Conference on Sensing Technology*, volume 1, 2005.

[20] Martin Maier, Finn-Niclas Stapelfeldt, and Vadim Issakov. Design approach of a k-band fmcw radar for breast cancer detection using a full system-level em simulation. In *2022 IEEE MTT-S International Microwave Biomedical Conference (IMBioC)*, pages 251–253. IEEE, 2022.

[21] Victor C Chen. *The micro-Doppler effect in radar*. Artech house, 2019.

[22] Avik Santra, Raghavendran Vagarappan Ulaganathan, Thomas Finke, Ashutosh Baheti, Dennis Noppeney, Jungmaier Reinhard Wolfgang, and Saverio Trotta. Short-range multi-mode continuous-wave radar for vital sign measurement and imaging. In *2018 IEEE Radar Conference (RadarConf18)*, pages 0946–0950. IEEE, 2018.

[23] Muhammad Arsalan, Avik Santra, and Christoph Will. Improved contactless heartbeat estimation in fmcw radar via kalman filter tracking. *IEEE Sensors Letters*, 4(5):1–4, 2020.

[24] Faheem Khan and Sung Ho Cho. A detailed algorithm for vital sign monitoring of a stationary/non-stationary human through ir-uwb radar. *Sensors*, 17(2):290, 2017.

[25] Qisong Wu, Zengyang Mei, Zhichao Lai, Dianze Li, and Dixian Zhao. A non-contact vital signs detection in a multi-channel 77ghz lfmcw radar system. *IEEE Access*, 9:49614–49628, 2021.

[26] Justin Saluja, Joaquin Casanova, and Jenshan Lin. A supervised machine learning algorithm for heart-rate detection using doppler motion-sensing radar. *IEEE Journal of Electromagnetics, RF and Microwaves in Medicine and Biology*, 4(1):45–51, 2019.

[27] Srikrishna Iyer, Leo Zhao, Manoj Prabhakar Mohan, Joe Jimeno, Mohammed Yakoob Siyal, Arokiaswami Alphones, and Muhammad Faeyz Karim. mm-wave radar-based vital signs monitoring and arrhythmia detection using machine learning. *Sensors*, 22(9):3106, 2022.

[28] Nebojša Malešević, Vladimir Petrović, Minja Belić, Christian Antfolk, Veljko Mihajlović, and Milica Janković. Contactless real-time heartbeat detection via 24 ghz continuous-wave doppler radar using artificial neural networks. *Sensors*, 20(8):2351, 2020.

[29] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169, 2021.

[30] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[31] Mostafa Alizadeh, George Shaker, João Carlos Martins De Almeida, Plinio Pelegrini Morita, and Safeddin Safavi-Naeini. Remote monitoring of human vital signs using mm-wave fmcw radar. *IEEE Access*, 7:54958–54968, 2019.

[32] Yong Wang, Wen Wang, Mu Zhou, Aihu Ren, and Zengshan Tian. Remote monitoring of human vital signs based on 77-ghz mm-wave fmcw radar. *Sensors*, 20(10):2999, 2020.

[33] Hyunjae Lee, Byung-Hyun Kim, Jin-Kwan Park, and Jong-Gwan Yook. A novel vital-sign sensing algorithm for multiple subjects based on 24-ghz fmcw doppler radar. *Remote Sensing*, 11(10):1237, 2019.

[34] Wenjie Lv, Wangdong He, Xipeng Lin, and Jungang Miao. Non-contact monitoring of human vital signs using fmcw millimeter wave radar in the 120 ghz band. *Sensors*, 21(8):2732, 2021.

[35] Jian Gong, Xinyu Zhang, Kaixin Lin, Ju Ren, Yaoxue Zhang, and Wenxun Qiu. Rf vital sign sensing under free body movement. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(3):1–22, 2021.

[36] Dingyang Wang, Sungwon Yoo, and Sung Ho Cho. Experimental comparison of ir-uwb radar and fmcw radar for vital signs. *Sensors*, 20(22):6695, 2020.

[37] SP Rana, M Dey, R Brown, HU Siddiqui, and S Dudley. Remote vital sign recognition through machine learning augmented uwb. In *IET Conference Proceedings*. The Institution of Engineering & Technology, 2018.

[38] Muhammad Imran Khan, Mian Ahmad Jan, Yar Muhammad, Dinh-Thuan Do, Constandinos X Mavromoustakis, Evangelos Pallis, et al. Tracking vital signs of a patient using channel state information and machine learning for a smart healthcare system. *Neural Computing and Applications*, pages 1–15, 2021.

[39] Xin Liu, Ziheng Jiang, Josh Fromm, Xuhai Xu, Shwetak Patel, and Daniel McDuff. Metaphys: few-shot adaptation for non-contact physiological measurement. In *Proceedings of the conference on health, inference, and learning*, pages 154–163, 2021.

[40] Infineon Technologies AG. Xensiv™ 60ghz radar sensor for advanced sensing, 2021. `https://www.infineon.com/cms/en/product/sensor/radar-sensors/radar-sensors-for-iot/60ghz-radar/bgt60tr13c/`, Last accessed on 2022-10-28.

[41] Vernier. Go direct® respiration belt, 2020. `https://www.vernier.com/product/go-direct-respiration-belt/`, Last accessed on 2022-11-08.

[42] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[43] James M Joyce. Kullback-leibler divergence. In *International encyclopedia of statistical science*, pages 720–722. Springer, 2011.

[44] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[45] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. *arXiv preprint arXiv:1810.09502*, 2018.

Chapter 8

Context-Adaptable Radar-Based People Counting via Few-Shot Learning

Gianfranco Mauro[1,2], Ignacio Martinez-Rodriguez[1], Julius Ott[1,3], Lorenzo Servadei[1,3], Robert Wille[3], Manuel P. Cuellar[4], Diego P. Morales[2].

1. Infineon Technologies AG, Am Campeon 1-15, 85579 Neubiberg, Germany
2. Department of Electronics and Computer Technology, University of Granada, 18071 Granada, Spain
3. Department of Electrical and Computer Engineering, Technical University of Munich, Arcisstrasse 21, 80333 Munich, Germany
4. Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain

Applied Intelligence Journal, Pages: 1-29, Springer Nature

- Received February 2023, Accepted June 2023, Published July 2023
- DOI: 10.1007/s10489-023-04778-z
- Impact factor (2022): 5.3
- JCR Rank (2022): 48/145 in category Computer Science, Artificial Intelligence (Q2)

Abstract.

In many industrial or healthcare contexts, keeping track of the number of people is essential. Radar systems, with their low overall cost and power consumption, enable privacy-friendly monitoring in many use cases. Yet, radar data are hard to interpret and incompatible with most computer vision strategies. Many current deep learning-based systems achieve high monitoring performance but are strongly context-dependent. In this work, we show how context generalization approaches can let the monitoring system fit unseen radar scenarios without adaptation steps. We collect data via a 60 GHz frequency-modulated continuous wave in three office rooms with up to three people and preprocess them in the frequency domain. Then, using meta learning, specifically the Weighting-Injection Net, we generate relationship scores between the few training datasets and query data. We further present an optimization-based approach coupled with weighting networks that can increase the training stability when only very few training examples are available. Finally, we use pool-based sampling active learning to fine-tune the model in new scenarios, labeling only the most uncertain data. Without adaptation needs, we achieve over 80 % and 70 % accuracy by testing the meta learning algorithms in new radar positions and a new office, respectively.

Keywords: active learning, meta learning, radar, few shot learning, people counting, weighting network.

8.1. Introduction

Counting the number of people in an environment can be a crucial task not only in industrial settings but also in medical and safety scenarios. In difficult times, such as during a pandemic, keeping track of the occupancy of an environment can greatly reduce the risk of spreading a pathogen [1, 2]. Estimating the presence of people can lead to other advantages, such as enabling energy management plans in places with frequent turnover of people, such as hospitals, by smartly activating equipment and heating systems [3]. A non-automated measure may be challenging or impossible in many contexts, such as for pedestrian crowds in public areas [4]. The majority of solutions designed for people monitoring rely on images captured by cameras and thermal sensors [5]. Most camera-based solutions use RGB or time of flight (ToF) sensors, and occupancy information is estimated using computer vision [6, 7] or machine learning [8, 9, 10]. Camera systems that use cross techniques for image segmentation and edge detection, such as convolutional neural networks (CNNs), achieve high performance even in crowded environments, but suffer from the inherent problem of a lack of privacy [11]. Thermal sensors, on the other hand, are much less privacy-invasive because of the usage of infrared frequencies and often lower image resolution [12]. Thermal sensors

also have the advantage of being usable in the dark, but they can be affected by thermal noise, caused, for example, by heaters and sunlight. In addition, the lack of depth information generally does not allow distinguishing between people moving in the same direction. In contrast to visual solutions, many other systems exploit the measurement of environmental quantities. Radio-frequency (RF) and laser technologies are typically classified as non-image-based approaches [13]. The CO_2 sensors, for example, can be used to estimate the occupancy of a room by the concentration of carbon dioxide produced by individuals. Such systems are frequently low-power but must account for venting systems and are practically unusable in open spaces [14]. LiDARs represent often another privacy-friendly solution for people counting and tracking. Through the use of pulsed lasers and a scanner, a LiDAR yields the generation of 2-D or 3-D maps of the surrounding space [15, 16]. Such systems frequently have high spatial resolution and frame rates, but they can be costly and power-consuming. RF-based systems have the advantage of having almost no privacy concerns and little dependence on light and weather conditions. These characteristics make them appropriate for monitoring several people. Wi-Fi technology, for example, can enable the recognition and segmentation of people even through walls and obstructions [17, 18]. Wi-Fi modules, however, require the development of high output power in the RF range ($\approx$ W) and a continuous working operation to exploit their functionalities. On the contrary, radar sensors are more versatile in many applications thanks to lower power consumption ($\approx$ mW) and optimized system power management. Among radar modulations, frequency-modulated continuous wave (FMCW) is particularly suited to people monitoring, allowing accurate estimation of the range and velocity of both dynamic and static targets located within the device's field of view (FoV) [19, 20]. Specifically, 60 GHz technology is particularly suitable for short-range people monitoring applications [21]. Radars transmitting around this frequency are cost-effective and versatile compared to other solutions such as cameras or LiDAR. Further, the 60 GHz frequency is much less susceptible to interference with other radio-frequency signals or Bluetooth devices. Image-based or high-resolution RF systems often implement a vision-based pipeline to predict the number of people in a given context. This approach can lead to high classification performance even in the challenging task of tracking through image segmentation, edge detection, and skeleton-pose extraction [6]. On the other hand, radar data are hardly interpretable through classical computer vision approaches. In this case, deep learning (DL) techniques are commonly used to process the information [22].

DL is nowadays finding the most varied uses for solving tasks and speeding up processes. Over the years, classes of DL models have been developed to extract valuable information from the available data for given tasks. Examples are CNNs for feature map generation or recurrent neural networks

(RNNs) for processing time series. Over the years, multiple neural network topologies, such as Inception [23] and VGGNet [24] have been designed to solve specific tasks with successful outcomes.Yet, such topologies have the inherent need to be trained on a large amount of data to achieve robust performance across new contexts. Commonly, these models are adaptable to new tasks by leveraging transfer learning [25], tailoring parameters to newly collected data. However, the limited availability of data and the need for rapid adaptation to new contexts make transfer learning hardly usable for defined types of tasks. To deal with these challenges, a specific branch of DL called few-shot learning has gained momentum in recent years [26]. The goal of few-shot learning is to exploit the little available information and data patterns, leveraging previous experience to adapt to new contexts or solve tasks that have not been tackled before.Few-shot learning is approached from different perspectives by specific DL sub-branches such as meta learning and active learning [27, 28].

Meta learning, or *learning to learn*, accounts for the set of algorithms where the primary goal is to learn how to approach new tasks given some past experience, or meta-data [29, 30]. This process not only encourages context generalization but also accelerates the fine-tuning of already observed tasks when new data are available. If the meta learning is optimization-based, an iterative learning process called episodic learning based on available training data is generally used. For a task defined in N–way, i.e., N classes, the few available samples are called shots. To assess generalization performance, C samples of *support* and J samples of *query* are fed to the defined model for each class. Algorithms commonly used for meta learning are model agnostic meta learning (MAML) [31] and Reptile [32] which, thanks to their very general conceptualization, enable the episodic adaptation of most of the common topologies defined in DL. Frameworks based on optimization-based meta learning are highly effective and perform well in several data-poor tasks [33, 34]. However, they have an inherent need for training on a set of representative data for each new, unseen task to learn to generalize. A specific kind of method, called relation network [35], was created to obviate this need by exploiting the ability of the model to compare the features of different examples and learn to distinguish them. The comparison is possible by properly shaping the model topology and regressing a relation score between 0 and 1, comparing individual support and query examples. The relation scores are unconventionally regressed by minimizing the mean squared error (MSE) to the ground truth of query instances. This approach assumes that all available support instances are mutually independent of each other. Intuitively, the model relies on a one-to-one comparison rather than comparing the new query examples with all the available support samples. Such issues are addressed by the weighting network [36]. In this adapted topology, the relation between support and query is propagated through two modules. A

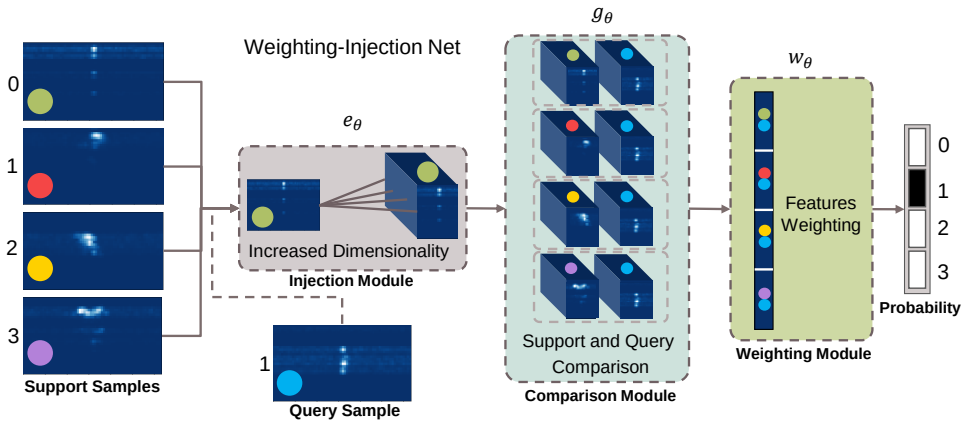


Figure 8.1: Weighting network with an injection module (Weighting-Injection Net). At least one instance per class, represented in the figure with a different marker color and a label, is used as support. A query example belonging to one of the classes is what is to be associated with a label by the classification algorithm. An injection module trained on the support images enables the concatenation of a query with an increased-dimensionality representation of each support. A comparison module merges support and query information by mapping the relation into a one-dimensional vector. Finally, a weighting module composed of fully connected layers maps the relational information to the query label. The model parameters are represented by θ.

first comparison module for the extraction of the similarity between the samples and a second weighting module that compresses the information into a one-dimensional vector representing the relation scores. This method leverages all available support sample features for query prediction. Further, the weighting network endorses the use of traditional classification cost functions such as crossentropy during episodic optimization.

Active learning, on the other hand, aims to optimize the model's performance with as few labeled instances as possible [37, 38]. To accomplish this, the algorithm has control over the inputs on which it trains, labels, or requests additional information about the data it deems most useful for learning. A common strategy is to assign a *priority score* to the unlabeled data pool, exploiting, for example, the probability distribution generated by the model. Only the instances identified as most uncertain are then labeled and used during training. This procedure, called pool-based sampling, is normally repeated multiple times, increasing the amount of labeled training data, until satisfactory performance for a given task is achieved.

In this paper, we exhibit how few-shot learning techniques can grant generalization of scenarios (environments and locations) for an FMCW radar-based algorithm designed for people counting. The application of this system is intended for uncrowded areas or rooms where there is a need to count the

presence of a few people. For this work, a specific dataset was collected using a 60 GHz radar that was set up for the task of counting people. The information was gathered in three different offices with at least four different in-room locations. Per location, 0 to 3 people took part in the data recording for at least 60 seconds per session. The data were preprocessed in frequency to extract range and Doppler information from the people in the scene. Meta learning is then used for the monitoring use case, estimating the number of people from radar data. Instead of using all the available data in a single training, we propose a few-shot episodic approach to foster and speed up adaptation. To meet the learning needs, we introduce both a new relation topology, which we call the Weighting-Injection Net, and an algorithm, which we call model-agnostic meta-weighting (MAMW). The Weighting-Injection Net represents a modification to the traditional weighting network presented in [36]. Instead of an embedding module that reduces the dimensionality of the support samples for the next comparison step, the proposed one uses an injection module. This module increases the dimensionality of input data, generating a feature-enriched representation of support and query samples for the next relational phase. The overall network scheme is shown in Figure 8.1. The MAMW, on the other hand, combines the query relation strategy of the weighted network with the two-step optimization-based approach of MAML. This is meant to improve the stability of the few-shot episodic training, especially when only very few instances are available as training. Experiments with 1–, 2–, 5–, and 10–shot have been performed and analyzed for the proposed methods. The achieved generalization results have been compared with those of other state-of-the-art approaches. State-of-the-art comparisons are also conducted up to five-person counting, to test the limitations of the radar-based episodic approach.

We also exhibit how pool-based sampling active learning can be efficiently employed to fine-tune the performance of a relational model by exploiting the most uncertain data. Showing how, for adaptations in new contexts, the use of generalization information learned from episodic adaptation leads to a better fit than starting from random initialization. The active learning strategy has been used to fit the 1–shot-pre-trained model on data from an office room used as a test that is therefore unseen in the meta-training phase.

For the meta learning algorithms, we also conducted experiments on a publicly available dataset for few-shot learning in the Appendix 8.8. The main contributions of this paper are as follows:

1. Implementation, to the best of our knowledge, of the first context-adaptable radar-based solution for counting people without a necessary adaptation training.

2. Design and implementation of the Weighting-Injection Net. This network represents a variation of the weighting network with an injection

module. The injection operation increases the dimension of support and queries to ease feature matching in the subsequent comparison module.

3. Design of a cross-algorithm between MAML and the weighting network, called MAMW to increase the training stability of 1– and 2–shot experiments.

4. Development of a pool-based sampling active learning algorithm compatible with weighting network topologies.

8.2. Related Works

In this section, we first investigate state-of-the-art solutions for people counting that offer similar features to radar-based systems, such as privacy preservation and low frame resolution. We then focus on the specific approaches aimed at context generalization and active learning.

When low frame resolution and privacy are system needs, traditional image segmentation and detection methods are often replaced or aided by deep learning. Neural networks can also be used to process time series or generate density maps for crowd monitoring.

Massa et al. [39] presented a recurrent neural network (RNN) architecture called LRCN-RetailNet (Long-term recurrent convolutional network) that takes as input sequences of low-resolution RGB frames and analyzes their spatiotemporal content for people counting. The strategy outperforms other state-of-the-art single-image-based approaches. The system based on temporal sequences may be unusable in low frame rate scenarios or with hardware implementation constraints. Gomez et al. [40] developed a system using long-wave infrared imaging and a CNN implementation on the NXP® LPC54102 microcontroller. The classification approach is binary, exploiting a small detection window on image sections to predict the presence or absence of heads. Because all weights fit in a 512 KB flash memory, the CNN can be easily deployed on the microcontroller. The counting algorithm using the embedded version of the model achieves an accuracy of 53.7 % on test images and up to six people. This solution is very low-power and privacy-friendly, but the presence of heat sources in the environment could cause counting issues due to the low resolution of the thermal sensor.

The most common types of RF-based systems used for monitoring are Wi-Fi and radars that use impulse radio ultra-wide band (IR-UWB) or FMCW technology. Most of these solutions are inherently characterized by privacy preservation and low sensor resolution. Kianoush et al. [41] presented a people counting system via Wi-Fi radio infrastructure that uses an ensemble of models to leverage the space-frequency features of various transmission and reception channels. The ensemble exploits Bayesian techniques based on

signal propagation statistics from RX to TX, a feed-forward neural network (FF-NN), and long-short-term memory (LSTM). Some of the constructed ensembles achieve an accuracy of over 95 % in the test setup. However, a network of Wi-Fi terminals is employed for this purpose, which results in higher power consumption and challenges usability in other environments. Bao et al. [42] featured a CNN-based algorithm for people counting focusing on extracting multi-scale range-time maps from IR-UWB radar data. Sequences of radar frames are preprocessed to extract the peak information and remove the background. The single frames are then stacked together to form range-time maps. The method proved robust in counting up to 10 people in the selected environment. However, the time dependency and lack of velocity information may make the system unsuitable for real-time applications where multiple people may be at the same distance. Stephan et al. [43] proposed a people counting solution via the *BGT60TR13* radar system (60 GHz FMCW) that makes use of knowledge distillation from synchronized camera data during the model generation. The suggested architecture first processes the camera RGB data, exploiting an OpenPose network that extracts the people's poses through pre-trained layers of the VGG-16 network and a multi-stage CNN. The extracted information is then fed to a triplet network with a 32-D embedding layer to generate clusters for each person count class. Radar information is first preprocessed in the form of range Doppler images (RDI) and fed to an encoder with fully connected final layers that learn through knowledge distillation from camera embeddings. Information transfer is possible by minimizing the Kullback-Leibler (KL) divergence between radar and camera embeddings. The method is robust and leads, in the test phase, to an accuracy of up to 71 % for six people with another radar sensor with different positions and orientations. What is learned through knowledge distillation, however, could significantly affect the capabilities of the architecture in new environments where morphological and light conditions would directly influence the camera data.

A few cutting-edge works attempt to solve the people counting problem through active learning or aim at context generalization.

Vandoni et al. [44] featured a solution that uses active learning, coupled with SVMs, to improve training on subareas of crowd images via head count. Samples that are more dissimilar than those already tagged are estimated in terms of their uncertainty via a metric that accounts for crowd density, called maximum excess over subarrays (MESA). Zhao et al. [45] also proposed an active learning solution for head counting in camera-based density maps. In this case, in the iterative process of instances sampling to be labeled, both crowd density information and dissimilarity from previous selections are employed. The sampling technique is a context-appropriate version of partition-based sample selection with weights (PSSW). The number of people is then regressed through mean absolute error (MAE) and MSE. Both

methods presented in [44] and [45] result effective in improving the people count through uncertainty sampling in crowded scenes but are very dependent on the 2D RGB nature of the images. Zhang Yingying et al. [46] proposed a multi-column convolutional neural network (MCNN) to estimate crowd head counts from single images without temporal dependence. Even with a sparse number of people, the method outperforms other cutting-edge solutions on a variety of public datasets. The model, trained on a large dataset with various density map sizes, can be easily tuned for new datasets and contexts via transfer learning. The required resolution is nonetheless high and could create context-specific privacy issues. Reddy et al. [47] and Zan et al. [48] designed an adaptive algorithm to generate crowd density maps using MAML with episodic training. In [47] a backbone consisting of the first layers of VGG-16 and a density map estimator are trained on various RGB sequences collected in different environments. The pioneering approaches depict how meta learning can be effectively employed for people counting. Hou X. et al. [49] presented a cross-domain solution for the estimation of density maps by episodic learning. In this case, a domain-invariant feature representation module is exploited, where synthetic and real camera data are used as source and target domains, respectively. The density maps are then generated using a pre-trained CNN network and an algorithm called β-MAML, where β represents the generalization step's learning rate. The parameter β is dynamically adapted in the episodes by exploiting the gradient information of parts of the images. The number of people is finally estimated from the density maps. The meta learning approach presents more robust performance for the algorithm than other state-of-the-art methods for density map generation. However, the need for a sensor camera does not allow for low-resolution uses or where privacy is a requirement.

Some cutting-edge RF-based works also propose adaptive context generalization solutions. Hou H. et al. [50] illustrated a few-shot learning solution for indoor crowd counting using Wi-Fi technology. The solution consists of a two-stage framework called domain-agnostic and sample-efficient wireless indoor crowd (DaseCount). In a first stage of meta-training, two separate CNNs learn to extract human activity information from wireless channel state information (CSI) measurements. Generalization performance is improved at this stage by knowledge distillation. In the meta-testing phase, the features extracted via CNNs from the CSI data are fed to a few-shot regression algorithm for the people counting task. The presented framework achieves, on average, over 96 % accuracy for counting up to eight people in various domain setups. Yet, the solution is computationally expensive for classifier retraining and may not be suitable for frequent Wi-Fi transceiver location changes. Zhang Yong et al. [51] proposed a WI-FI-based few-shot learning solution for activity recognition that makes use of graph neural networks. The method uses a graph convolutional block attention module

to extract activity-related information from CSI data. A final classification layer is used to classify the graph features and recognize the activity. The approach presents a robust 99.74 % accuracy in the 5–way 5–shot experiment for new environments and activities. Yet, much computation and memory are required for model adaptations.

8.3. System Setup and Radar Preprocessing

In this section, we propose a general overview of the system, discuss the data acquisition setup, and provide information about the employed radar board, its configuration, and the main preprocessing steps.

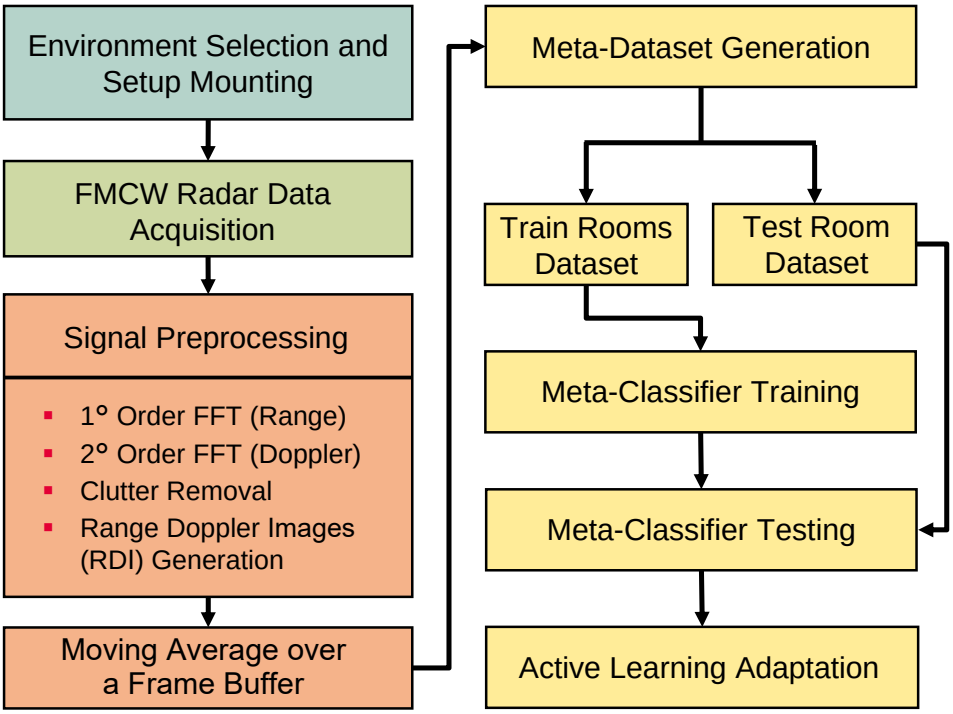


Figure 8.2: Proposed Framework. The setup is mounted in three rooms. Data sessions with a number of people from 0 to 3 in the scenario are collected and processed (orange). The frequency analysis is performed via the fast Fourier transform (FFT). Instances are generated via a moving average over frame sequences. A meta-dataset is then generated, and one room is used as the test dataset. A classifier is then episodically trained and tested. Active learning is used to fine-tune the model to a new environment (yellow).

8.3.1. General Overview of the System

Figure 8.2 depicts the overall framework. First, rooms for data gathering are chosen for the few-shot learning approach. The radar data are then gathered from various in-room locations with varying numbers of people. Preprocessing is performed to extract range and Doppler information about the people in the FoV of the device. The sequences of preprocessed frames are averaged by moving average to generate the individual instances of the meta-dataset. The data are then saved and labeled in session-specific folders. The folder names denote the label encoding, from 0 to 3, of the number of people who attended the session. In most of the proposed experiments, the information recorded in two rooms is used as input data for the episodic training of the meta learning model. The third room is instead utilized for testing. Model fine-tuning can be performed via active learning on the test data, using the meta learning model as a baseline.

8.3.2. Radar Board

All radar data in this work were collected using the *BGT60TR13C* FMCW sensor [21] from Infineon Technologies AG. With a center frequency of f_0 of 60 GHz and a bandwidth of about 6 GHz, this radar represents a miniaturized and low-power solution. This f_0 and bandwidth are especially suitable in short-distance and indoor applications, resulting in low susceptibility to interference with other signals such as WiFi or Bluetooth. Thanks to an operation-optimized duty cycle, the power consumption for sensing within 5 m is minimized to only 5 mW. The *BGT60TR13C* has a transmit (TX) and three receive (RX) channels built into the package. The RX antennas are placed orthogonally to each other to enable the reconstruction of azimuth and elevation angles of arrival (AoA) for the targets placed in the FoV. The information collected from the RX channels is mixed with the TX and digitized with 12-bit resolution via the board connected to the radar sensor (Figure 8.3).

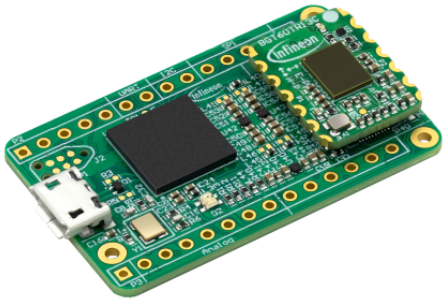


Figure 8.3: *BGT60TR13* Radar System. The board filters, mixes, and digitizes data from each RX channel, located on top of the radar sensor.

8.3.3. Radar Configuration

The *BGT60TR13C* transmits a series of linearly frequency-modulated signals called chirps in a defined bandwidth B_w around the central frequency f_0. Each chirp, of duration t_c, normally consists of a fixed number of samples n_s. During use, the information reflected in the RX channels is mixed with a transmitted signal reference and digitized, thus generating an output signal called intermediate frequency (IF). Normally, for further preprocessing, the radar information is packed into frames, each containing the IF relative to a sequence of chirps N_c. The theoretical maximum detection range R_{max} and range resolution Δr of an FMCW modulation are calculated using the following formulas:

$$\Delta r = \frac{c}{2B_w} \ , \tag{8.1}$$

$$R_{max} = \frac{\Delta r}{2} n_s \ , \tag{8.2}$$

where c stands for the speed of light in air. A narrow B_w of 0.48 GHz was chosen to achieve a R_{max} of about 10 m, which would cover the entire size of the chosen environments. A resolution Δr of at least 31 cm was chosen to let several targets placed in front of the radar be distinguished even at a considerable distance. A n_s per chirp of 64 has been specifically selected. The maximum discernible velocity of the targets V_{max} in one direction and the resolution Δv can instead be calculated with the following formulas:

$$V_{max} = \frac{c}{4 f_0 t_c} \ , \tag{8.3}$$

$$\Delta v = \frac{2V_{max}}{N_c}. \tag{8.4}$$

The average human walking speed is about 1.42 m/s. To allow detecting even faster motions, we opted for a V_{max} of 3.5 m/s and a Δv of 1.1 cm/s. As a result, we set t_c to 351 μs and N_c to 64. To collect approximately seven frames every half second, a frame repetition time fps of 75 ms was chosen. Furthermore, an analog-to-digital converter (ADC) sampling rate F_s of 2 MHz was chosen. The parameters used to configure the *BGT60TR13C* for the people counting recordings in all the selected rooms are listed in Table 8.1.

8.3.4. Recording Setup

The *BGT60TR13C* radar system was mounted on a tripod for the people counting data, and the data were collected using a Raspberry® Pi 4. The raw

Table 8.1: Radar Sensor Parameters Configuration.

Symbol	Quantity	Value
f_0	center frequency	60 GHz
fps	frames per second	13.33
N_c	number of chirps	64
n_s	samples per chirp	64
t_c	chirp time duration	351 µs
B_w	bandwidth	[59.78 – 60.26] GHz
F_s	sampling frequency ADC	2 MHz

radar data were then processed and labeled offline at a later time on an eight-generation Intel® Core™ i5 processor (4 cores). Figure 8.4 depicts the used setup. Three different rooms of various sizes were chosen for data collection: an office of approximately 26 m^2 and two meeting rooms of about 20 and 39 m^2, respectively. Only a portion of the office has been used, with walls separating the other two areas. Various types of furniture, such as cabinets, desks, tables, and chairs, were left in the rooms and were unmoved from their locations. The reflection of such objects represents the so-called clutter that characterizes the FMCW radar data. A graphical illustration of the three environments, indicated with the letters S, M, and B, standing for small, medium, and big, is provided in Figure 8.5. Data were gathered in each room from at least the four corners. Data were also collected in three additional locations in the office room. At every location, the tripod was set up at a height ranging from 1.65 to 1.75 meters. Four sessions have been carried out per location, each lasting approximately 60 seconds for the meeting rooms and 90 seconds for the office. Each session contains data from 0 up to a maximum of 3 people in the room at the same time. Ten different people with heights ranging from 1.60 to 1.78 meters took part in the recordings. Some data up to 5 people have been gathered in the big room to further test the performance of the developed algorithm. Before collecting data, user consent was obtained, and as much privacy and data anonymization as possible were maintained during the recordings. The collected data has not been made publicly available.

8.3.5. Radar Preprocessing

Raw radar frames are difficult to interpret and label. The information to be fed to a DL model for learning purposes can be too noisy and highly context-dependent due to clutter. In this work, we propose to preprocess the raw data collected for people counting by removing the clutter and extracting the Doppler and range information of the targets through frequency analysis with the fast Fourier transform (FFT). We then perform two averages to

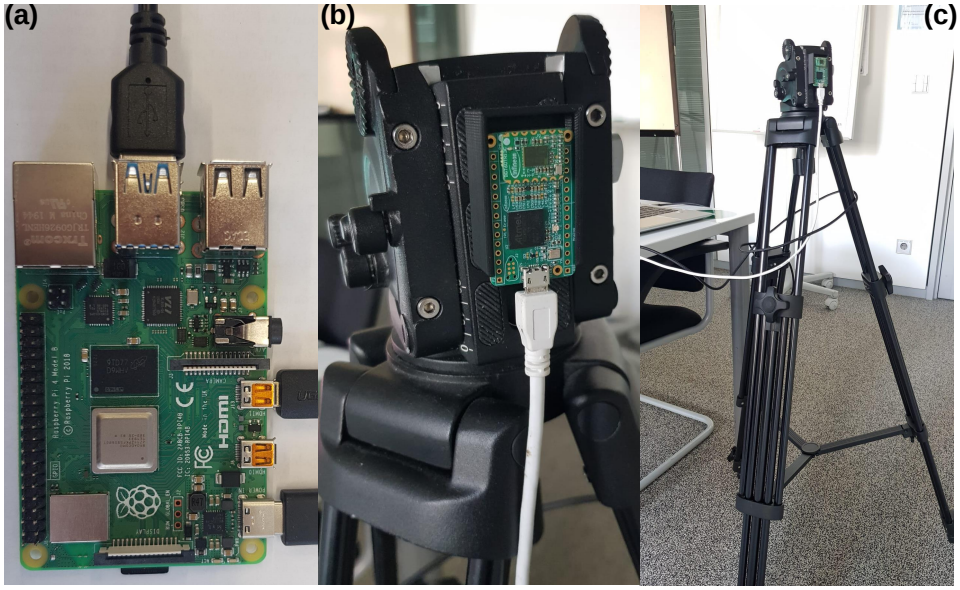


Figure 8.4: Data recording setup. A Raspberry® Pi4 (**a**) is used for data storage. For data collection, the *BGT60TR13C* radar system is mounted on the tripod (**b**). The tripod is moved between sessions in the various rooms and locations (**c**).

reduce the noise in the data for the next model generation step. One for each frame, averaging the IF signal $Ch_{IF}(i)$ generated for each of the three RX channels ($i \in I_{RX}$), and another for each 7-frame recorded series. The whole process, given the fps of 75 ms, leads to the generation of about 2 RDI per second. The main preprocessing steps are shown in Figure 8.6.

The preprocessing steps performed for each RX-generated IF signal are as follows:

1. For each chirp (slow time), the average value of the samples (fast time) is calculated and then subtracted.

2. The IF signal is then multiplied in fast time with a Hanning window to reduce the spectral leakage effects.

3. A 1-D FFT is performed on the samples to derive the range information of the targets.

4. A multiplication with a Hanning window is run also in the slow time.

5. A 1-D FFT is performed along the slow time to obtain the velocity information.

6. To drop the information of static objects, aka clutter, moving target

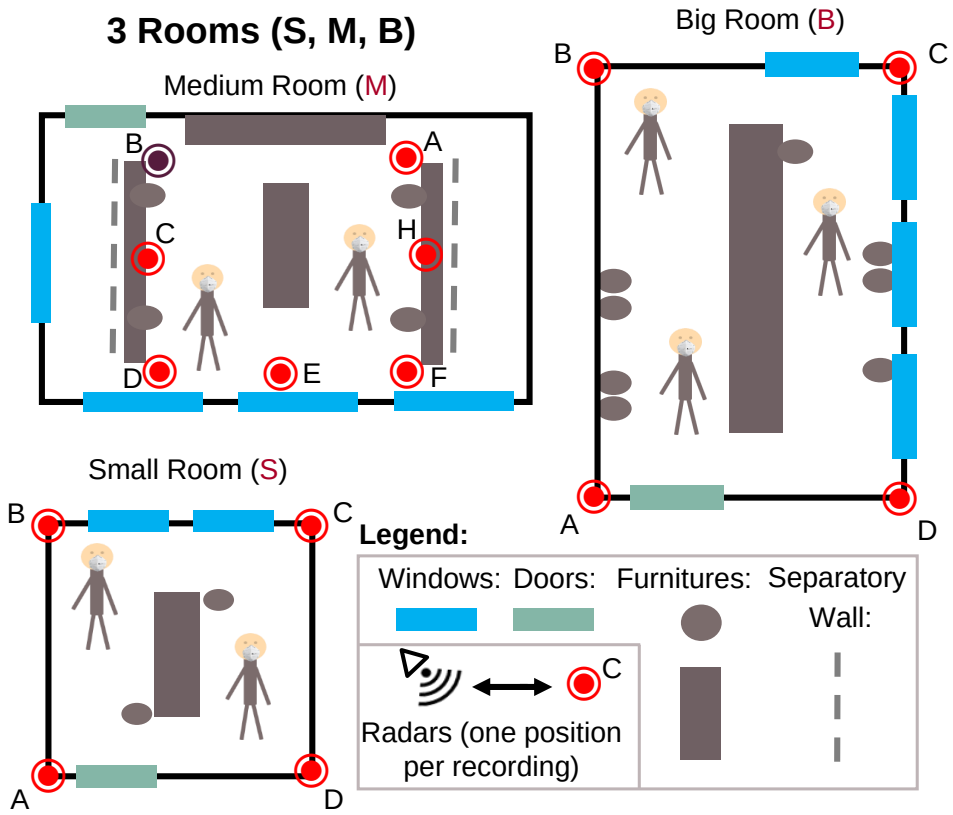


Figure 8.5: A graphic illustration of the environments chosen for data collection. Data from 0 to 3 people were collected from the four corners of the rooms. For the office M, data were also gathered at three other locations (C, E, and H, respectively). For M, data could not be collected from location B due to the presence of the front door.

indication (MTI) is utilized (Equation 8.5).

$$Ch_{IF}(i) = \mu Ch_{IF}(i) + (1-\mu)\overline{Ch_{IF}}(i) \ , \tag{8.5}$$

where $\mu \in [0, 1]$ is set to 0.9, and weights the importance of the current frame against the average of the previous ones $\overline{Ch_{IF}}(i)$.

7. For each $Ch_{IF}(i)$ a constant false alarm rate (CFAR) algorithm is used to locally select Range and Doppler peaks in frequency and discard the surrounding information, thus increasing the signal-to-noise ratio (SNR).

8. To further improve the SNR, the $RDIs(v)$ for each frame $v \in V$ are computed as the absolute value of the average of $Ch_{IF}(i)$ (Equa-

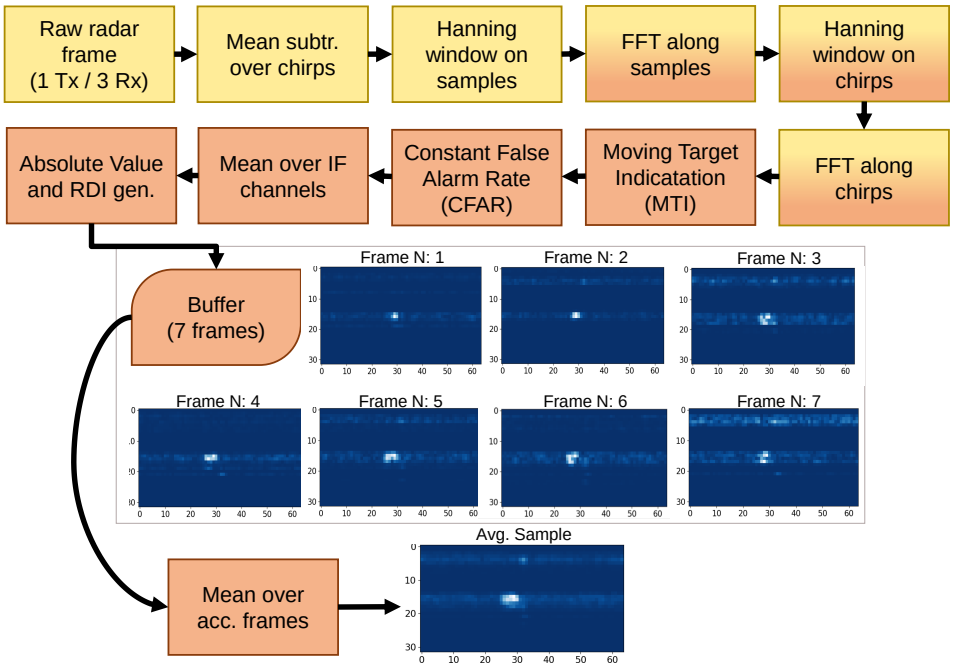


Figure 8.6: Flow diagram representing the main preprocessing steps. The yellow blocks represent the main time-domain steps. The orange ones instead represent the frequency domain steps.

tion 8.6).

$$RDI(v) = \left| \frac{1}{I_{RX}} \sum_{n=0}^{I_{RX}} Ch_{IF}(i) \right| . \tag{8.6}$$

9. The $RDIs$ thus generated are stored in a seven frames buffer (N_v), which corresponds to roughly half the frame rate. A moving average is performed on the buffer to further reduce the noise in the $RDIs$. These $RDIs$ represent the individual instances of the people counting dataset that get labeled (Equation 8.7).

$$RDI = \left| \frac{1}{N_v} \sum_{v=0}^{N_v} RDI(v) \right| . \tag{8.7}$$

8.3.6. People Counting Dataset

For people counting, three different meta-datasets have been generated from the collected data of up to three people. Given a frame timing of 75 ms and the frames averaged performed on a seven frames buffer, a total of 7,669 labeled samples have been created. Each sample has a size of 32 times 64 pixels. The width of 64 pixels represents the velocity span, corresponding to

the number of chirps per frame. The height of 32 pixels represents the range span, corresponding to half of the bin samples per frame. Independently of the recording room, labels represents the number of people P_m in the recording, with $m \in [0, 3]$. As shown in Figure 8.5, the data has been divided into sub-folders of the tuple (R, P_m, and L). The tuple components are the room's name R: S, M, or B, the number of people (P_m), and the location, $L \in [A, H]$. With an average duration of 60 seconds across all recordings in rooms S and B, a total of 1,677 and 1,702 examples were created, respectively. For M, a total of 4,290 examples were built with six available locations. With all the available instances, the following three meta-datasets have been generated:

- *Mixed-Dataset*: the data from the sub-folders (R, P_m, L) were randomly split so that approximately 75 % of the instances was training and 25 % was testing. The number of training and test instances in this case are 5,803 and 1,866, respectively.

- *S-Test-Dataset*: in this case, all sub-folders (S, P_m, L) were used as tests, while all others ($[M, B]$, P_m, L) were used as training. In total, for this meta-dataset, there are 5,922 training examples and 1,677 test examples.

- *B-Test-Dataset*: all the sub-folders (B, P_m, L) were used as test, while all the others ($[S, M]$, P_m, L) were used as training. The number of training and test instances are 5,967 and 1,702, respectively.

In general, for each of the three generated meta-datasets, the training and test instances are part of the respective training $\mathcal{D}^{m-train}$ and test $\mathcal{D}^{m-test}$ meta-dataset splits. Three different averaged RDI examples per class, sampled from the different recordings in all rooms and locations, are shown in Figure 8.7.

Even in the same environment, RDIs from classes 1 to 3 are difficult to distinguish from one another. Figure 8.8 shows a t-distributed stochastic neighbor embedding (t-SNE) with a 2-D component representation of all instances in the S room. The t-SNE succeeds in correctly clustering only data with zero people in the environment. A t-SNE representation of all collected data are shown in Figure 8.9 according to the *B-Test-Dataset* split. Even with a larger amount of data, only the zero-person instances are easily clustered. In this case, it can also be observed that the test data, which represents the B room, have different features than the rest of the points. This is an important indication of the dependence of radar data on the location in which they are collected. Algorithms trained in a single location may be difficult to use in other environments and usually require adaptation. Euclidean distance was used as a metric, and Barnes-Hut was used as an optimization algorithm to generate the t-SNE representation.

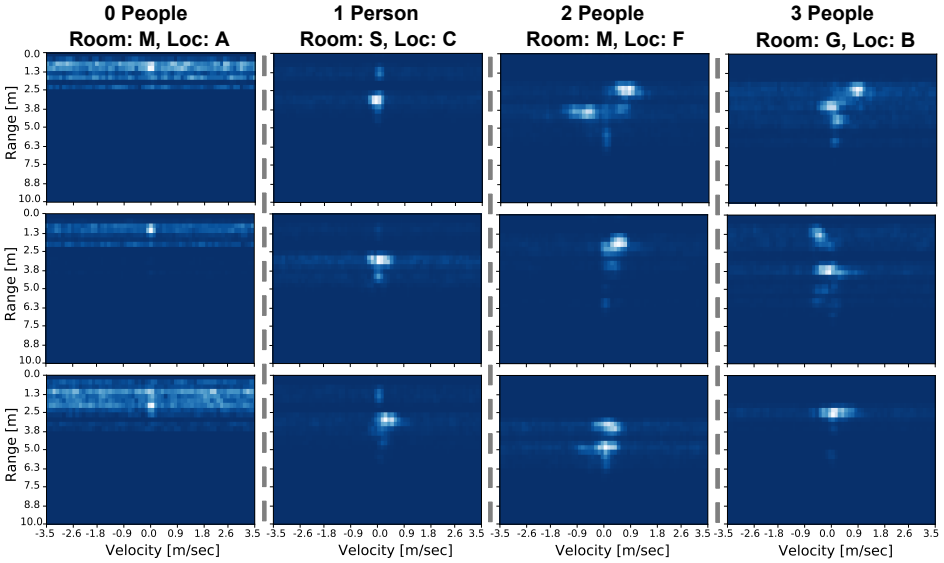


Figure 8.7: Example RDI instances from the people counting dataset. Every row shows three examples per class, chosen from a random combination of rooms and locations. The axes indicate people relative motion velocity in m/sec and distance from the radar sensor in cm.

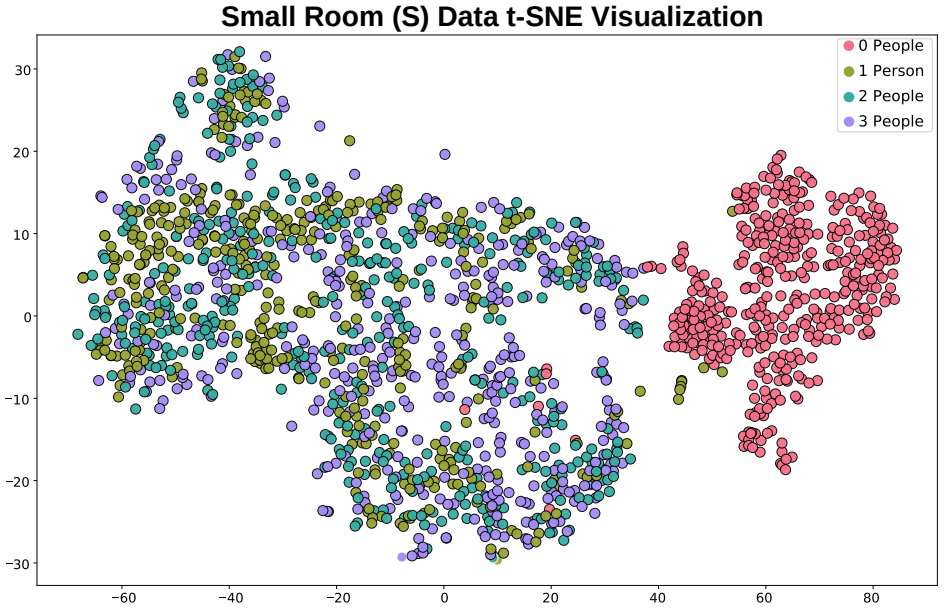


Figure 8.8: 2-D t-SNE representation of all S room data. This t-SNE was obtained with a perplexity of 40 over 6,000 optimization iterations.

8.4. Proposed Approach

In this section, we present our solutions for generalization learning. We begin by proposing a new network topology called the Weighting-Injection

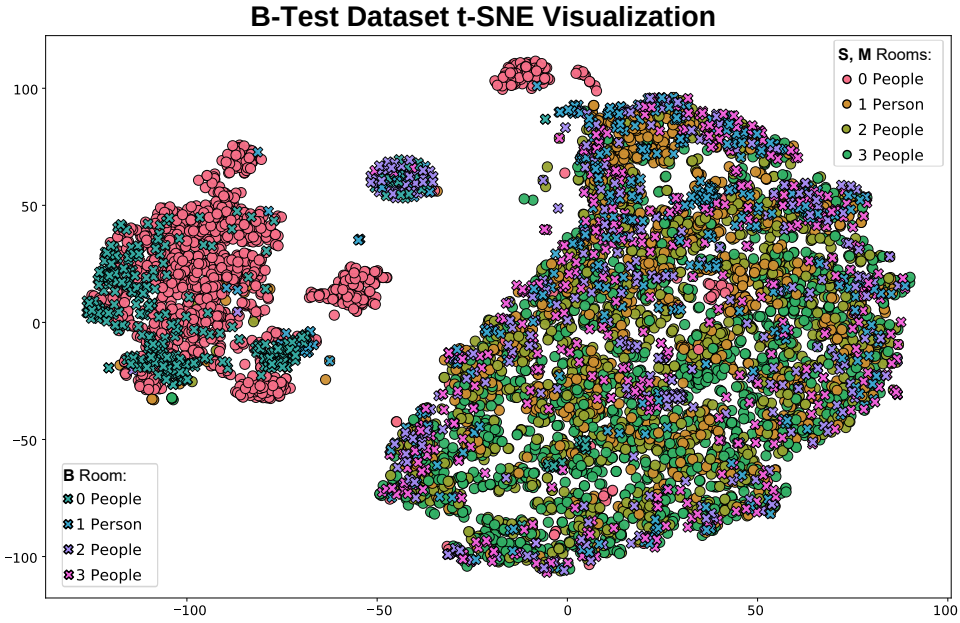


Figure 8.9: 2-D t-SNE representation of the *B-Test-Dataset*, for all the recorded data. The B room data are represented by the "x"marker, while the rest of the data (rooms S and M) are represented by the .º"marker. This representation was obtained with a perplexity of 30 over 7,000 optimization iterations.

Net, which is inspired by the weighting network [36]. We then propose an algorithm that makes use of optimization-based meta learning features from MAML [31], which we call MAMW. This modified version aims at increasing training stability when only a very limited number of shots per class are available. Then, we propose an active learning strategy tailored for weighting networks to allow fine-tuning in a new environment while minimizing the amount of required labeled data.

8.4.1. Meta Learning

In episodic meta learning, K tasks are sampled from a distribution $p(\mathcal{T}_r)$ defined over $\mathcal{D}^{m-train}$. As the episodes progress, the goal is to improve the performance of the model on tasks sampled from $p(\mathcal{T}_s)$ defined on $\mathcal{D}^{m-test}$. In DL, task-based learning is often achieved via the gradient method, which involves training the parameters θ' by minimizing a cost function $\mathcal{L}_{\mathcal{T}_r}(f_{\theta'})$, where $f_{\theta'}$ represents the relation between the input x and the predicted output $\hat{y}$. In the relation networks [35], generalization among tasks is directly achieved thanks to the intrinsic comparison of instances enabled by the topology. In optimization-based meta learning, such as in MAML [31], the information learned for tasks $\mathcal{T}_r$ and encoded in the parameters θ', is

transferred to a base model f_θ with parameters θ, minimizing an outer cost function $\mathcal{L}_{\mathcal{T}_r}(f_{\theta'})$. In this case, the task-specific cost function depends on the parameters θ of the base model $\mathcal{L}_{\mathcal{T}_r}(f_\theta)$.

8.4.1.1. Weighting-Injection Net

The Weighting-Injection Net aims to compare the features of the arbitrary examples of query q with those of reference to the support s classes for each task $k \in K$. The Weighting-Injection Net, as shown in Figure 8.1 is based on three main modules: injection, comparison, and weighting. During training, the gradient information is propagated through all modules in both forward and backpropagation steps. For a N–way 1–shot task, the idea is to map the relationship between support examples s_n, where $n \in \mathbb{N}$: [1, 2, ..., N], to each query example q_j, where j is the index of the j-th example of the set.

The injection module e_θ generates a higher dimension representation of the input x to enhance the extraction and matching of features in the subsequent comparison step. Gradient information for the injection module is only propagated as $e_{\theta'}(s_n)$ through the support instances. For the query, only the feature representation $e_{\theta'}(q_j)$ is generated.

The comparison module c_θ, takes as input the concatenation along N channels of $e_{\theta'}(q_j)$, with each of the n support samples. The number of channels N corresponds to the task number of ways. The features are extracted in the module using convolution layer sequences, yielding a comparison vector z. The vector z is generated in the following way:

$$z_{n,j} = g_{\theta'}(e_{\theta'}(s_n) \parallel e_{\theta'}(q_j)) \ , \tag{8.8}$$

where $\parallel$ denotes the operation of concatenation along the N channels.

Lastly, the weighting module w_θ is designed to generate a probability density from the concatenated N channels in the z vector. Each $z_{n,j}$ is the output of the comparison module, between the query q_j and a support s_n. The predicted output $\hat{y}_j$ for the sample q_j can be expressed as follows:

$$\hat{y}_j = w_{\theta'}(\bigm|\bigm|_{n=1}^{N} z_{n,j}) = w_{\theta'}(z_{1,j} \parallel z_{2,j} \cdots \parallel z_{N,j}) \ , \tag{8.9}$$

where $||$ represents the sequence of concatenations performed over the channels N of z.

In the case of a N-way C–shot task, where $c \in \mathbb{N}$: [1, 2, ..., C], the supports per class can be denoted as $s_{n,c}$. The Weighting-Injection Net can be leveraged in this case to create a more robust representation of the comparison vector $z_{n,j}$. This can be done by arithmetic averaging over C sets of N-channel concatenations, given by the embedded representations of q_j

with each of the support sets $s_{n,c}$. Such a more robust representation yields the query class estimation with less bias than with the single support shot scenario. The mathematical expression for a single q_j is as follows:

$$z_{n,j} = \frac{1}{C} \sum_{c=1}^{C} g_{\theta'}(e_{\theta'}(s_{n,c}) \parallel e_{\theta'}(q_j)) \ . \tag{8.10}$$

The Weighting-Injection Net, trained on $p(\mathcal{T}_r)$, can be tested, thanks to its inherent structure, on tasks from $p(\mathcal{T}_s)$ without further training. Given a support set with elements $s_{n,c}$ for a task $\mathcal{T} \sim p(\mathcal{T}_s)$ a N–way C–shot, the class probability density of the j-th query sample q_j, is directly estimated by inference.

8.4.1.2. Model-Agnostic Meta-Weighting

The weighting network [36] represents a robust episodic learning algorithm thanks to the inherent feature of instance comparison. Yet, this method can be characterized by learning instability when only a few-shot per class are available. Especially in 1–shot learning, this is due to the comparison of the query with the individual support instances, which may not be sufficiently descriptive of a class for a given task. Hence, we present a method called model-agnostic meta-weighting (MAMW), which tries to incorporate within the weighting network some features of optimization-based meta learning to enhance the stability and robustness of prediction in this setting. Specifically, in the MAMW, we propose to divide episodic learning into inner and outer steps. Given a N–way C–shot task:

1. In the *inner step*, the support instances are compared with a noisy version of themselves of Gaussian type via a function $e_\theta(\phi((s_{n,c})))$. This noise is generated at random from the $\mathcal{N}(0, \sigma^2)$ distribution in the interval $[-\sigma, \sigma]$. Defined s_h as the h-th support example, where $H = N \cdot C \implies h \in \mathbb{N}$: [1, 2, ..., H], the computation of $z_{n,h}$ can be expressed as follows:

$$z_{n,h} = \frac{1}{C} \sum_{c=1}^{C} g_\theta(e_\theta(s_{n,c}) \parallel e_\theta(\phi(s_h))) \ , \tag{8.11}$$

$$\hat{y}_h = w_\theta(\mathop{\Big\|}_{n=1}^{N} z_{n,h}) \ , \tag{8.12}$$

where θ represent the parameters of the base model f_θ. Such operations can also be carried out in batches. An example of people counting instances compared with their noisy version is shown in Figure 8.10.

2. In the *outer step*, the comparison between the support examples $s_{n,c}$ and each query q_j is performed, starting from the weights θ' learned in the inner loop. In this case, the comparison vectors z are computed with the Equation (8.10) and the predicted output $\hat{y}_j$ with Equation (8.9).

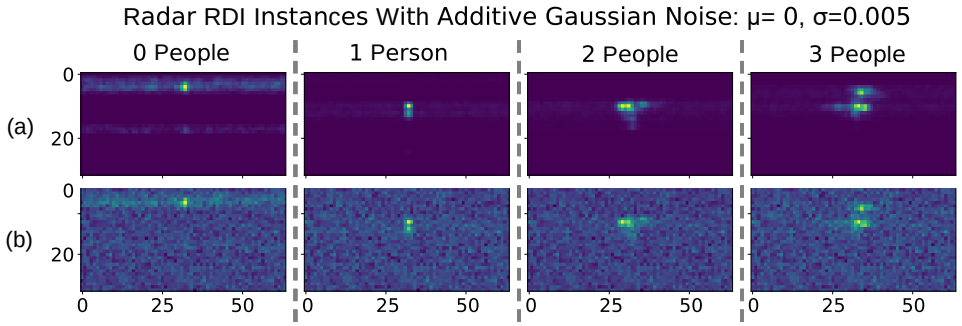


Figure 8.10: Examples of RDI without (**a**) and with added Gaussian noise (**b**) used in the inner step training of the MAMW.

The main steps of the MAMW, in the case of few-shot, supervised learning with outer updates after every task, are defined in Algorithm 1.

The presented Weighting-Injection Net topology can be trained via the MAMW algorithm. Also with the MAMW episodic learning, the Weighting-Injection Net can tackle new test tasks without the necessary adaptation training. MAMW does not need algorithmic modifications when an embedding module is used instead of the injection module.

8.4.2. Active Learning

Active learning can also be used on top of a meta learning model to perform fine-tuning on a given task, leveraging the most uncertain queries during adaptation. We propose to use pool-based sampling active learning to fine-tune the Weighting-Injection Net on $p(\mathcal{T}_s)$, starting from what has been learned on $p(\mathcal{T}_r)$. We chose an uncertainty sampling strategy to let the algorithm decide at each training epoch which new examples to label. We test the approach with three different priority scores: least confidence (LC), margin sampling (MS), and entropy (E), respectively. For the instances $q_j = \{x_j,\ y_j\}$ representing the input/output pairs on queries sampled by $\mathcal{T}$, the priority scores S_p can be defined as follows:

$$S_{LC} = \underset{x_j}{\operatorname{argmax}}\ (1 - P_\theta(\hat{y}_{max} \mid x_j))\ , \tag{8.13}$$

$$S_{MS} = \underset{x_j}{\operatorname{argmin}}\ (P_\theta(\hat{y}_{max} \mid x_j) - P_\theta(\hat{y}_{max-1} \mid x_j))\ , \tag{8.14}$$

Algorithm 1 MAMW for N–way C–shot Supervised Learning

Ensure: N–way: $n \in \mathbb{N}$: [1, 2, ..., N]
Ensure: C–shot: $c \in \mathbb{N}$: [1, 2, ..., C]
Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
Randomly initialize θ
Random sample K tasks $\mathcal{T}$ from $p(\mathcal{T})$
for $\mathcal{T}_k \in \mathcal{T}$ **do**
    Sample $H = N \cdot C$ support instances s_h from $\mathcal{T}_k$
    for all s_h **do**
        Compute $z_{n,h}$ in Equation (8.11)
        Compute $\hat{y}_h$ in Equation (8.12)
        Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_k}(\hat{y}_h)$ by $\mathcal{L}_{\mathcal{T}_k}$ for s_h
        Compute adapted parameters with gradient descent: $\theta' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_k}(\hat{y}_h)$
    end for
    Sample J query instances q_j from $\mathcal{T}_k$
    Compute $z_{n,j}$ in Equation (8.10)
    Compute $\hat{y}_j$ in Equation (8.9)
    Update $\theta \leftarrow \theta - \beta \nabla_\theta \mathcal{L}_{\mathcal{T}_k}(\hat{y}_j)$ for q_j
end for

$$S_E = \underset{x_j}{\operatorname{argmax}} \left(- \sum_{n=1}^{N} P_\theta(\hat{y}_n \mid x_j) \log P_\theta(\hat{y}_n \mid x_j) \right) , \tag{8.15}$$

where P_θ of $\hat{y}_{max}$ is the highest posterior probability predicted by the model with θ parameters for x_j, and N is the number of classes.

Algorithm 2 defines the main step of the proposed pool-based sampling on a task $\mathcal{T}$. In general, the Algorithm 2 represents a generalization of the pool-based sampling approach for relational models. For a given task, a set of class-related support examples is initially labeled. As the number of iterations increases, the uncertainty of the query examples is evaluated, and those with the highest priority score are added to the labeled dataset. A maximum number of support instances per class per iteration is also chosen. Instead of starting with random weights, parameters learned during episodic learning on training tasks can be used as the model initialization. The active learning procedure is therefore performed on unseen test tasks.

8.5. Experimental Setup

In this section, we present all the results achieved on meta learning episodic experiments and active learning fine-tuning on the people counting meta-datasets (Section 8.3.6). The algorithms have been written in the Python

Algorithm 2 Pool-based Sampling Active Learning for N–way C–shot Supervised Learning on Weighting-Injection Net

Ensure: N–way: $n \in \mathbb{N}$: [1, 2, ..., N]
Ensure: C–shot: $c \in \mathbb{N}$: [1, 2, ..., C]
Require: Task $\mathcal{T} \sim p(\mathcal{T})$
Require: J: Queries to sample per epoch
Require: A: Queries to label per epoch
Initialize θ with meta-learned weights
Initialize $\mathcal{D}_p = \{\}$ as labeled Pool.
Sample in $\mathcal{T}$ support instances:
$s_{n,c} = \{x_{n,c},\ y_{n,c}\}$
Add all $s_{n,c}$ in $\mathcal{D}_p$
Sample in $\mathcal{T}$, J query instances:
$q_j = \{x_j,\ y_j\}$
while not done **do**
    Compute $z_{n,j}$ in Equation (8.10)
    Compute $\hat{y}_j$ in Equation (8.9)
    Compute S_p of q_j with Equation (8.13), (8.14) or (8.15)
    With S_p of q_j, select A queries q_{j_a} and $\hat{y}_{j_a}$
    Add all q_{j_a} in $\mathcal{D}_p$
    Update $\theta \leftarrow \theta - \beta \nabla_\theta \mathcal{L}_{\mathcal{T}_k}(\hat{y}_{j_a})$
    Sample in $\mathcal{D}_p$ support instances:
$s_{n,c} = \{x_{n,c},\ y_{n,c}\}$
    Sample in $\mathcal{D}_p$, J query instances:
$q_j = \{x_j,\ y_j\}$
end while

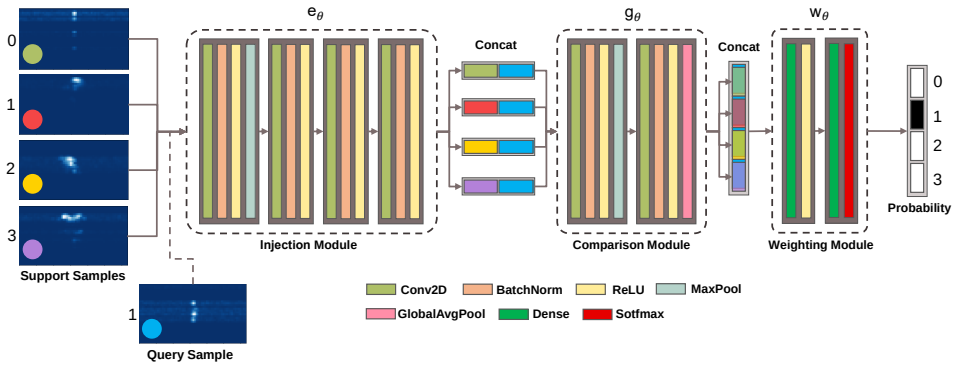


Figure 8.11: Representation of the topology modules and respective layers used in the relational experiments. The injection module (e_θ) increases the data dimensionality via a sequence of convolutional layers. The query sample is compared with all the available support samples. To combine relevant features, the comparison module (g_θ) employs convolution and global average pooling. The weighting module (w_θ) generates a feature matching probability density using dense layers and softmax activation.

programming language, using the TensorFlow™ module to implement the DL models. Further experiments on a public dataset have been performed and discussed in the Appendix 8.8. The codes related to the algorithms and topologies used for the meta learning experiments are available online[1]. As a process unit, we used an Nvidia® Tesla® P4 GPU and CUDA® Toolkit v11.1.0 for parallel computing.

8.5.1. Meta Learning Experiments

All the episodic experiments have been performed with the topology presented in Section 8.4.1.1 and Figure 8.1. Specifically, 4–way experiments with 1–, 2–, 5–, and 10–shot have been performed. The topology has been trained with two different algorithms. First with the classical episodic few-shot training of weighting networks, as defined in [36], using the Weighting-Injection Net equations (Section 8.4.1.1). Further, the topology has been trained in episodic sequences of inner and outer steps, following the steps of the MAMW algorithm proposed in Section 8.4.1.2. All the results presented in this section refer to the two algorithms and are consistently called Weighting-Injection Net and MAMW. Comparison results of the two algorithms with the state-of-the-art are presented in the Section 8.5.1.1. The cutting-edge comparison also features some application limit experiments for indoor people counting up to five individuals in a room.

[1]The codes for the meta learning algorithms are available at `https://github.com/GiancoMauro/TF-Meta-Learning/`

A graphical representation of the model modules and respective layers is shown in Figure 8.11. The model consists of 283,379 trainable parameters in its entire module sequence. Of the total, the injection module consists of 239,680 parameters, the comparison module of 39,936, and the weighting module of the remaining 4,180. To rescale feature size, max pooling is used in cascade to the 2D convolution (Conv2D) for the two modules e_θ and g_θ. In addition, batch normalization is used to increase the stability of training. All batch normalization layers are followed by a rectified linear unit (ReLU) activation function. To map the output vector into a probability distribution over the classes, the softmax is used as an activation function for w_θ. The cost function chosen for the query classification is categorical crossentropy, and the optimization algorithm is Adam. β_1 and β_2 for Adam have been set to 0 and 0.5, respectively. A learning rate of $5e-4$ has been chosen for the Weighting-Injection Net. A learning rate of $5e-4$ has also been chosen for both the inner and outer steps of MAMW. For the Gaussian noise statistic on the MAMW inner step, a value of σ^2 equal to 0.005 has been chosen. This value represents an empirical choice, noting that larger values led to the loss of the main information in the support instances, while smaller values were less effective for the performance of the experiments.

Regardless of the number of shots, every meta-training experiment is performed over 22,000 episodes, each of a single training epoch. The episodic learning is carried out on $\mathcal{D}^{m-train}$. The validation and testing have been performed at the end of each episode on 10-shot per class (40 samples) on tasks sampled by $\mathcal{D}^{m-train}$ and $\mathcal{D}^{m-test}$ respectively.

All experiments have been carried out with an embedding size g of 64. Smaller embedding sizes resulted in non-convergent experiments, whereas larger sizes resulted in meta-overfitting on $\mathcal{D}^{m-train}$. For the injection module, an output representation of $14 \cdot 14 \cdot g$ has been chosen (feature size). This led to a representation per image of 12,544 units. On the Nvidia® Tesla® P4 GPU, the number of floating points operations per second (FLOPS) for the injection module with this configuration is 108 megaFLOPS. The size in bytes of the weights of the model when saved in ".h5"format, regardless of the chosen episodic training algorithm and the number of shots, is 1,148 KB. Some experiments at varying feature sizes are also presented later in this section to test the benefits of the injection module over the standard embedding module.

The obtained values of prediction accuracy, model size, and single-sample prediction latency are compared to state-of-the-art values obtained by training other algorithms on the people counting dataset employed in this work. The accuracy results for the Weighting-Injection Net are reported for varying numbers of shots. Each experiment by algorithm, meta-dataset, and number of shots has been performed three times and tested on 10,000 final tasks sampled by $\mathcal{D}^{m-test}$. All presented results include the 95 % confidence

interval in addition to the average accuracy value.

Table 8.2: Network Layers Configuration - People Counting.

Module	**Type**	**Filter Shape**[1]	**Output Shape**
Injection	Conv2D	3×7×1×64	j×30×58×64
	MaxPool	2×2	j×15×29×64
	Conv2D	3×7×64×64	j×13×23×64
	Conv2D[2]	3×7×64×64	j×13×19×64
	Conv2D[2]	3×7×64×64	j×14×14×g
Comparison	Conv2D[2]	3×1×2g×g	jc×44×16×g
	MaxPool	3×3	jc×14×5×g
	Conv2D	3×3×g×g	jc×12×3×g
	AvgPool	1×1	jc×g
Weighting	Dense	ng×16	j×16
	Dense	1×n	j×n

The performance evaluation of each individual experiment is measured according to the validation and test accuracy values obtained by the model as the number of episodes increases. For every experiment, a box plot on the validation and testing accuracy statistics of tasks sampled by $\mathcal{D}^{m-train}$ and $\mathcal{D}^{m-test}$ is constructed every 2,200 episodes. In the following plots and paragraphs, statistical insights from one of the experiments performed are analyzed. Specifically, a MAMW 10–shot experiment on *Mixed-Dataset* is chosen thanks to the good achieved generalization performance. Figure 8.12 shows the set of box plots generated as the training episodes advance for the considered experiment. As the episodes progress, the mean and median values of the distributions rise while the quartiles and whiskers narrow. With episodes progressing, even the outliers move closer to the upper limit of accuracy. The described behavior demonstrates how, thanks to previously acquired experience, the model can generalize better on new sampled tasks. This means that newly learned parameters θ generalize better in new contexts, i.e., new locations and test rooms, resulting in higher performance under the same learning conditions.

Discrete accuracy density histograms can be used to represent the distribution underlying individual box plots. Graphical evidence of how the distribution tends to shift towards higher generalization accuracy can be observed by comparing the first and last histograms of the episodic optimization. Such density histograms can also be compared to a Gaussian probability distribution, thus showing what percentage of the achieved accuracy lies between the first and third quartiles. Figure 8.13 depicts a comparison of accuracy statistics for the examined experiment at the beginning and end of the episodic training. Even for tasks sampled only by $\mathcal{D}^{m-test}$, the probability density

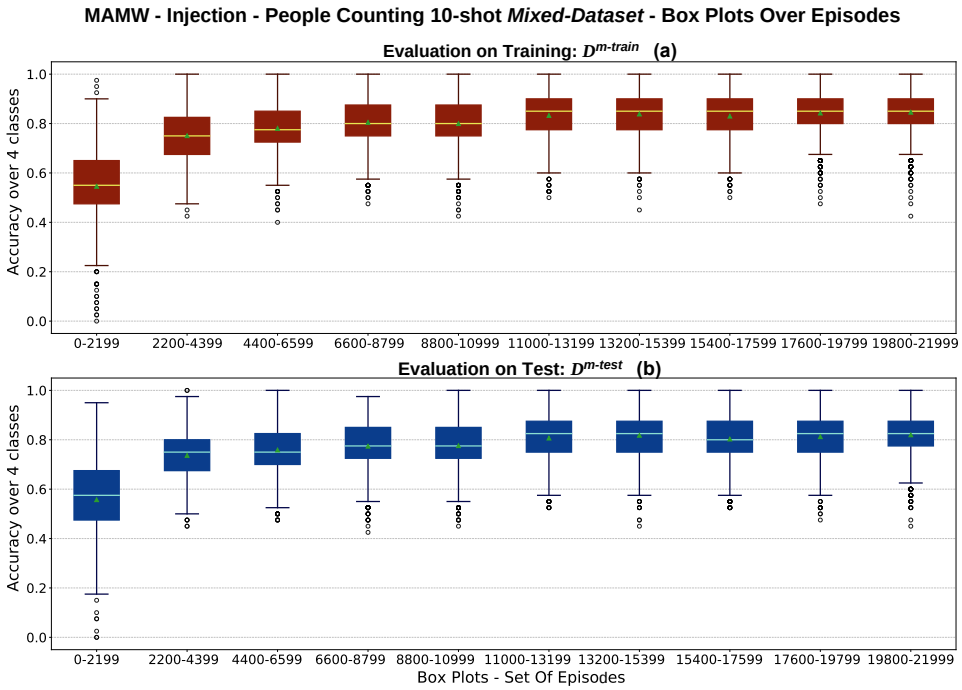


Figure 8.12: Accuracy statistics box plots vs. episodes for a MAMW 10–shot *Mixed-Dataset* experiment. The red box plots are generated on validation tasks(**a**), whereas the blue ones (**b**) are generated on test tasks. The median and mean values are represented by a horizontal line and a green triangle in each box plot. The small circles represent the box plot outliers.

tends, as the episodes progress, to take on a negative skew towards the upper limit of accuracy. The actual distributions underlying the box plots are not Gaussian but multi-modal with density peaks due to the variable complexity of the sampled tasks.

The generalization capability can be addressed at the level of individual classes by constructing cumulative confusion matrices on task sequences. Labels 0 to 3 represent the real and predicted number of people for the two dataset splits. Figure 8.14 depicts the confusion matrices underlying the first and last box plots of Figure 8.12 for both $\mathcal{D}^{m-train}$ and $\mathcal{D}^{m-test}$.

Figure 8.15 shows another example of cumulative confusion matrices for a Weighting-Injection Net 5–shot experiment on *S-Test-Dataset*. It is noticeable in both Figure 8.14 and Figure 8.15, that the model learns to generalize better as episodes progress for both unseen locations and rooms. Most misclassifications, especially at the end of episodic learning, lie around the main diagonal. This means that the models, in most cases, count ± 1 person compared to the actual number of individuals in the environment. Moreover, the majority of the misclassifications happen for the classes of 1 to 3 persons,

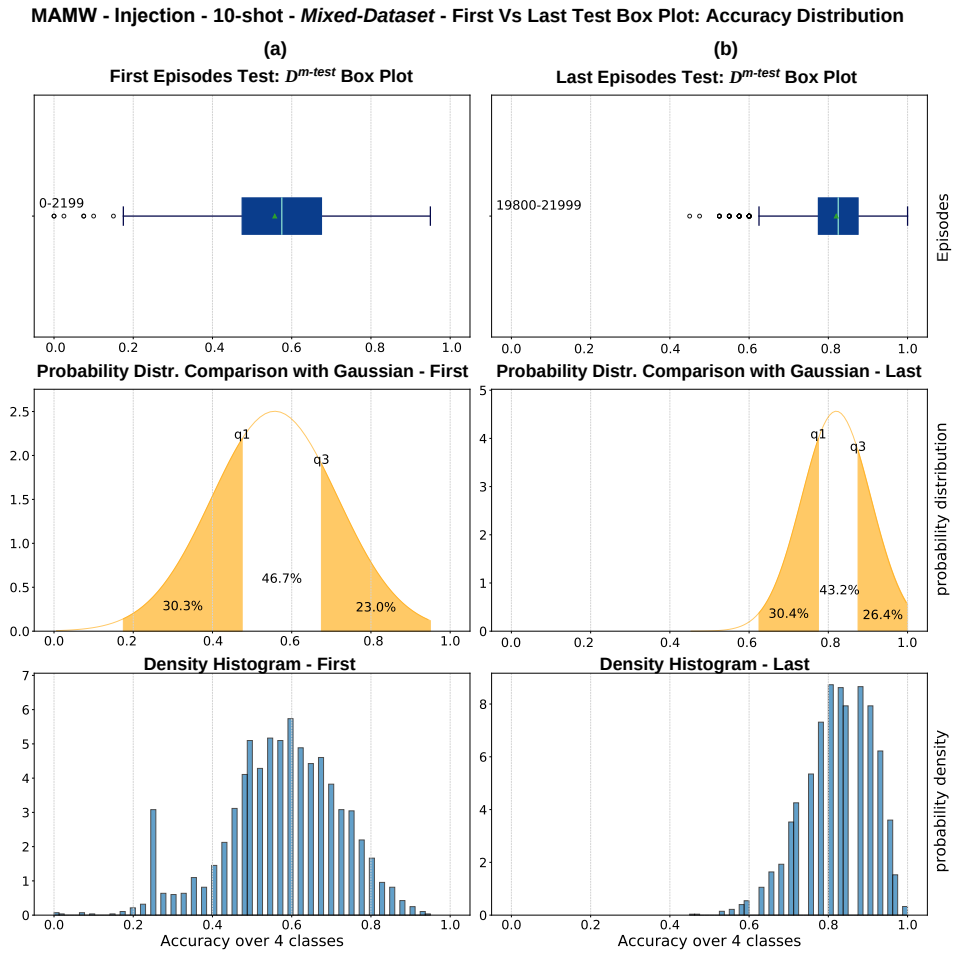


Figure 8.13: MAMW 10–shot experiment, first (**a**) and last (**b**) box plot underlying distributions, generated on test tasks sampled from *Mixed-Dataset*. The q1 and q2 values on the Gaussians indicate the first and third quartiles, respectively. The probability density histograms show the actual non-Gaussian nature of the distribution. The accuracy probability density for the last box plot (**b**) exhibits a negative skew as a result of the generalization learning.

while the model easily succeeds in distinguishing the case 0 that corresponds to no people detected in the sensor's FoV. The per-class accuracy of the test confusion matrices in Figure 8.15 turns out to be lower than that in Figure 8.14. This is due not only to the use of 10–shot instead of 5–shot in the experiment but also to the higher complexity of the test tasks. In fact, the Figure 8.15 experiment sampled all test tasks from a room not included in the training (S).

The prediction accuracy values obtained as an average of the post-training

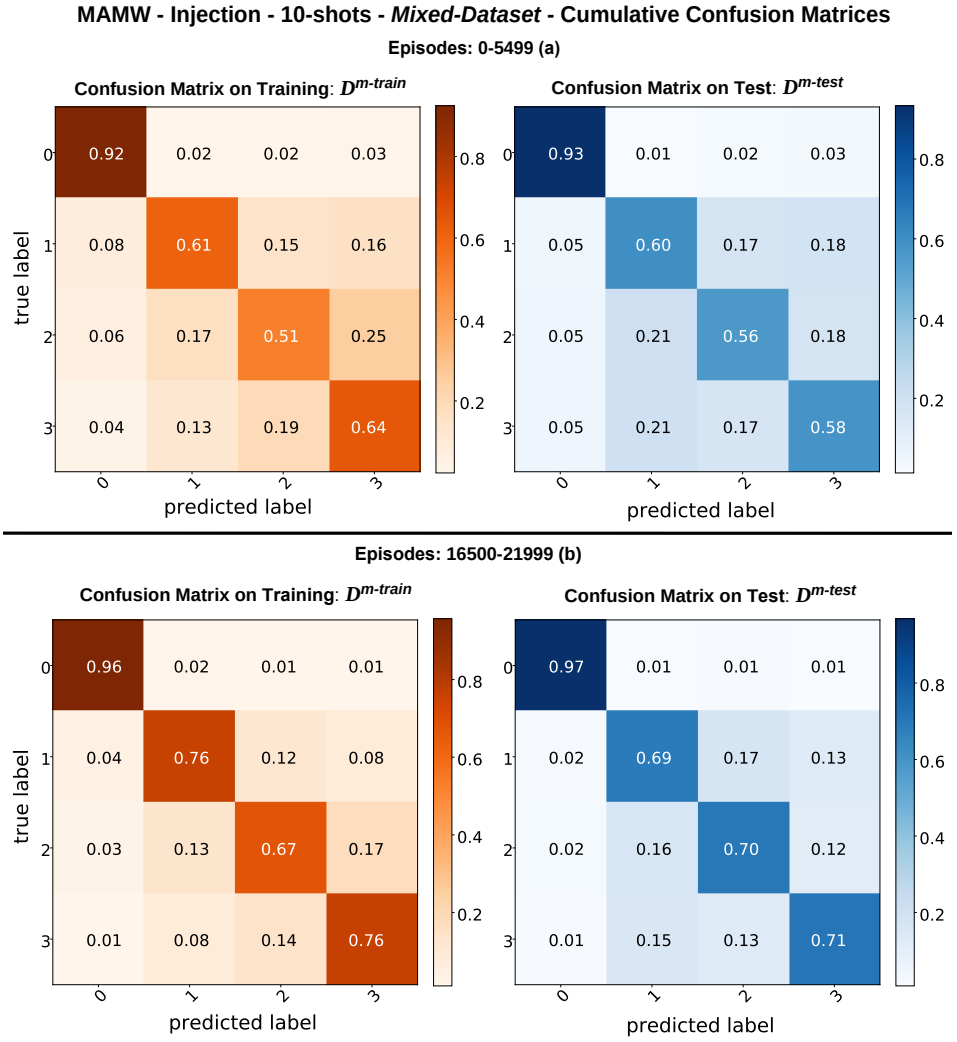


Figure 8.14: Cumulative confusion matrices for a 10–shot MAMW *Mixed-Dataset* experiment. Confusion matrices are obtained on the first (**a**) and last (**b**) 5,550 meta-iterations in the validation phase for both $\mathcal{D}^{m-train}$ and $\mathcal{D}^{m-test}$ sampled tasks.

tests for each experiment type are listed in Tables 8.3, 8.4, 8.5 for the three defined meta-datasets.

As can be observed from Tables 8.3, 8.4 and 8.5, regardless of the used meta-dataset, the 1– or 2–shot experiments performed with the MAMW lead to higher average accuracy values than the Weighting-Injection Net. In these specific cases, in episodic learning, the few supports per class make the prediction given by the Weighting-Injection Net less robust, where the learning depends solely and exclusively on the comparison with the query. MAMW instead supplies more information to the model thanks to the initial

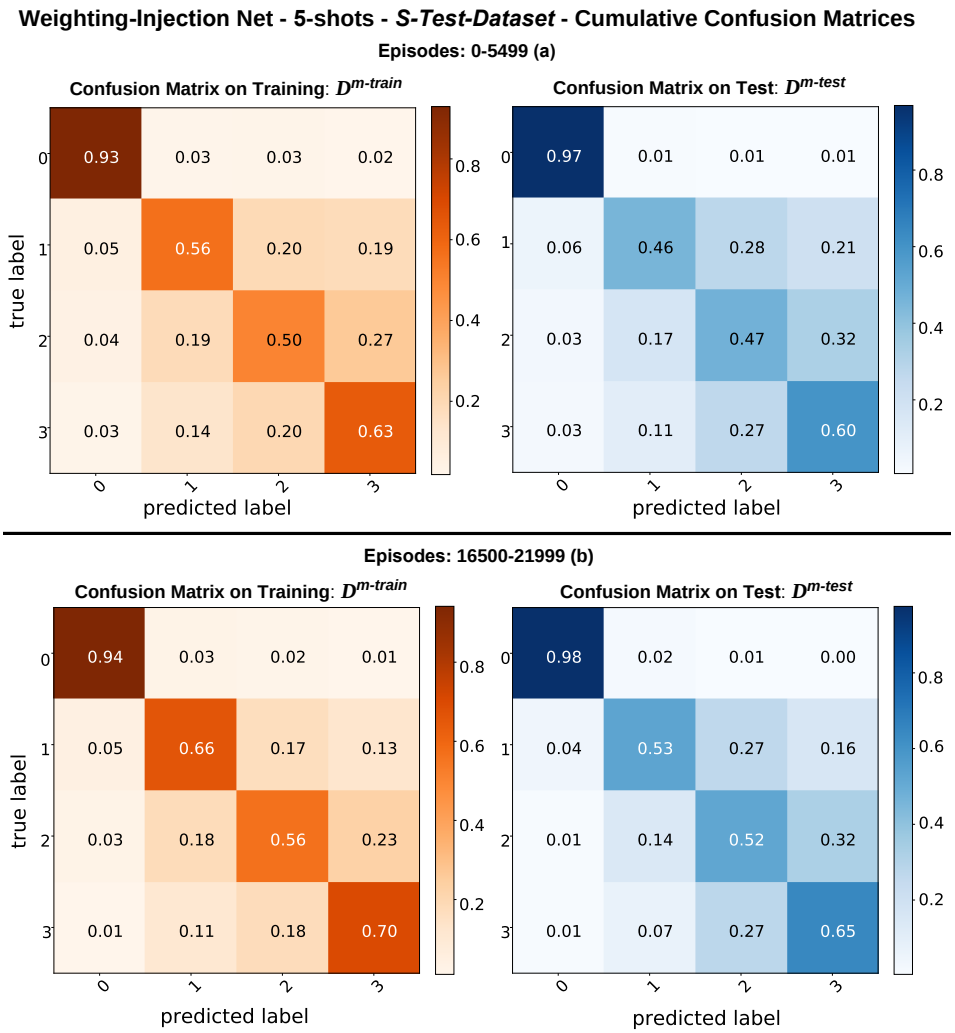


Figure 8.15: Cumulative confusion matrices for a 5–shot Weighting-Injection Net *S-Test-Dataset* experiment. Confusion matrices are obtained on the first (**a**) and last (**b**) 5,550 meta-iterations in the validation phase for both $\mathcal{D}^{m-train}$ and $\mathcal{D}^{m-test}$ sampled tasks. In this case, the entire S room is utilized as the test set.

comparison with a noisy version of the support samples, thus emphasizing the potential intrinsic noise of the query data. For the 5– and 10–shot experiments, the two episodic approaches lead to different performances with respect to the used meta-dataset. The MAMW outperforms the Weighting-Injection Net on the *Mixed-Dataset*, regardless of the number of shots. The *Mixed-Dataset* contains, in fact, recordings from all rooms, but with different locations and numbers of people. In this case, the MAMW goal of capturing noise similarity between support and query also aids query class recogni-

Table 8.3: Accuracy of the two meta learning approaches on people counting (4 classes): *Mixed-Dataset*.

Accuracy [%] ***Mixed-Dataset***	**Weighting-Injection Net**	**MAMW**
1–shot	63.01 ± 0.21	66.95 ± 0.22
2–shot	71.79 ± 0.20	74.10 ± 0.20
5–shot	78.26 ± 0.18	78.63 ± 0.19
10–shot	81.40 ± 0.16	82.16 ± 0.16

Table 8.4: Accuracy of the two meta learning approaches on people counting (4 classes): *S-Test-Dataset*.

Accuracy [%] ***S-Test-Dataset***	**Weighting-Injection Net**	**MAMW**
1–shot	59.85 ± 0.19	61.98 ± 0.19
2–shot	61.14 ± 0.16	64.48 ± 0.17
5–shot	71.77 ± 0.17	73.40 ± 0.18
10–shot	76.61 ± 0.16	73.53 ± 0.16

Table 8.5: Accuracy of the two meta learning approaches on people counting (4 classes): *B-Test-Dataset*.

Accuracy [%] ***B-Test-Dataset***	**Weighting-Injection Net**	**MAMW**
1–shot	54.26 ± 0.23	57.35 ± 0.23
2–shot	60.00 ± 0.22	60.83 ± 0.22
5–shot	69.98 ± 0.18	68.57 ± 0.18
10–shot	71.06 ± 0.18	70.74 ± 0.18

tion. This is thanks to the intrinsic features of the RDIs collected in the same room, which are thus influenced by the properties of that environment. On *S-Test-Dataset* and *B-Test-Dataset* instead, the Weighting-Injection Net outperforms MAMW in most 5– and 10–shot experiments. In these cases, given the relevant difference in context for the test room, the MAMW comparison with the noisy version of supports might shift the learning objective towards detecting noise rather than the class of query samples.

For relation-based topologies, there is no need to perform adaptation training for new tasks as a result of the direct comparison of features between the newly available support samples and the query. Therefore, the adaptation time to a new task is null. Instead, the inference time on a single sample

(query) can be computed as a function of the number of shots. It corresponds to the time required by the model to predict the query class given the available supports. The time required to compute the z comparison vectors for all available supports is thus included in the inference time for single queries. As both the proposed algorithms share the same inference procedure, these values are independent of the employed approach. The single sample inference time is also independent of the selected counting meta-dataset, given the same input size. Average inference values on a single query are listed in Table 8.6.

Table 8.6: Average single-sample inference time computed as the average of all MAMW and Weighting-Injection Net experiments on all defined meta-datasets, in function of the number of shots. Every experiment has been run over 10,000 final tasks on Nvidia® Tesla® P4 GPU.

Number of Shots	**Avg. Inference Time [ms]**
1–shot	14.46
2–shot	16.21
5–shot	27.03
10–shot	43.73

As can be seen from Table 8.6, the inference time for a single query increases as the number of shots increases. Multiple supports available per class enable a more robust prediction of the query class, as shown in Equation 8.10. However, this requires the generation of multiple z comparison vectors, which, in proportion to the number of shots, lead to a progressive increase in inference time on a single query.

Table 8.7: Accuracy achieved for the Weighting-Injection Net with varying feature size on people counting (4 classes): *S-Test-Dataset*. The chosen embedding size g is 64.

Accuracy [%] ***S-Test-Dataset***	**1,024** $(4 \cdot 4 \cdot g)$	**5,184** $(9 \cdot 9 \cdot g)$	**12,544** $(14 \cdot 14 \cdot g)$
1–shot	61.63 ± 0.20	60.17 ± 0.21	59.85 ± 0.19
2–shot	63.84 ± 0.18	63.83 ± 0.17	61.14 ± 0.16
5–shot	68.82 ± 0.18	68.63 ± 0.16	71.77 ± 0.17
10–shot	67.94 ± 0.16	71.49 ± 0.17	76.61 ± 0.16

Classification accuracy is also dependent on the chosen feature representation dimension in the feature extraction module e_θ. In specific experimental settings, the injection can counter episodic overfitting effects by increasing

Table 8.8: Mean classification accuracy achieved by the various algorithms, for experiments on people counting (4 classes): *S-Test-Dataset*.

Accuracy [%] *S-Test Dataset*	**Reptile**	**MAML 2nd**	**MAML+**	**Weighting-Injection Net**	**MAMW**
1–shot	49.61 ± 0.16	49.92 ± 0.18	52.53 ± 0.17	59.85 ± 0.19	61.98 ± 0.19
2–shot	52.02 ± 0.15	53.79 ± 0.16	56.91 ± 0.16	61.14 ± 0.16	64.48 ± 0.17
5–shot	57.95 ± 0.15	60.26 ± 0.17	60.38 ± 0.16	71.77 ± 0.17	73.40 ± 0.18
10–shot	63.00 ± 0.16	65.49 ± 0.17	64.67 ± 0.16	76.61 ± 0.16	73.53 ± 0.16

feature size as opposed to the standard embedding. The $14 \cdot 14$ feature size chosen for all the other experiments is compared with two representations of $4 \cdot 4$ and $9 \cdot 9$ respectively. Given the size of an RDI example of $32 \cdot 64 = 2,048$, a feature representation of $4 \cdot 4 \cdot 64 = 1,024$ converts the injection module into an embedding module. Compared with the 108 MegaFLOPS required by the feature size of $14 \cdot 14$, the size $4 \cdot 4$ requires only 0.28 MegaFLOPS. Overall, the injection operation, compared to embedding, results in the GPU performing significantly more FLOPS. This is due to the larger size of the extracted features in the convolutional layers.

Table 8.7 features the results on the *S-Test-Dataset*, obtained with the Weighting-Injection Net as feature size, and the number of shots vary. The 1–shot experiment seems to benefit more from embedding than from an injection module. The squeezed representation of features in such experiments leads to a more compact representation. The entire weighting network can succeed in extracting key features from the few samples available per class in each episode bringing benefits of generalized learning. On the other hand, as the number of shots increases, a larger representation of features seems to lead to greater benefits in training. With 5– or 10–shot per class, a larger feature space upstream of the comparison module facilitates feature extraction from the available support samples and yields better generalization results. The effect of overfitting on individual tasks is clearly visible by comparing the accuracy obtained with the $4 \cdot 4$ feature size between the 5– and 10–shot experiments. Contrary to the common scenario, the performance of the model worsens as the number of shots doubles. Without tuning the other hyperparameters, the small feature size favors single-task adaptation rather than generalized learning, reducing so, the overall performance.

8.5.1.1. Comparison with the State-of-the-Art and Limitations

In this section, the results of Weighting-Injection Net and MAMW are compared to the results of other state-of-the-art meta learning methods for the task of people counting. Reptile [32] is used as a baseline algorithm.

MAML 2^{nd} [31] and a more stabilized and performant version of MAML presented in Antoniou et al. [52], are the other algorithms used for comparison. The latter, labeled MAML$^+$, leverages the contributions of multi-step loss optimization (MSL), derivative-order annealing (DA), and cosine annealing of meta-optimizer learning rate (CA). The model chosen for the state-of-the-art algorithms is a CNN suitable for the generalization goal, consisting of four main blocks. The first three blocks consist of a Conv2D with 64, 128, and 256 filters, followed by batch normalization and the ReLU activation function. The last block consists of a dense layer with 4 neurons, corresponding to the number of classes. This topology consists of 403,332 trainable parameters compared to the 283,379 of MAMW and the Weighting-Injection Net. The adaptation training was done with Adam as the optimizer, with learning rates of $8e-3$ and $7e-3$ in the inner and outer cycles, respectively. Likewise, in this case, the values of β_1 and β_2 for Adam have been set to 0 and 0.5, respectively. The model training was executed on 22,000 episodes with a batch size of 2 and a number of epochs per task of 4, respectively. The comparison was performed on 10,000 final tasks on *S-Test-Dataset* for 1–, 2–, 5– and 10–shot over 3 repetitions of each experiment. For each task, 10 test samples per class were randomly selected, resulting in 40 test instances in total. The computed mean classification accuracy values are listed in Table 8.8. As can be observed, the MAMW turns out to be the best-performing method in all experiments apart from the 10–shot experiment, where, as commented in Section 8.5.1, the Weighting-Injection Net achieves a higher average accuracy. The accuracy values obtained with the proposed methods are better despite using 30 % fewer trainable parameters. As the number of shots increases, relation-based models show an even larger accuracy gap than optimization-based ones due to the more robust prediction given by averaging the comparison vectors computed for the available support samples.

Because of the direct mapping between sample and label in the learning process, the single-sample inference time for Reptile, MAML 2^{nd} and MAML$^+$ is independent of the number of shots. Across all the experiments, on an average of 10,000 final tasks, the overall estimated inference time has been 33.47 ms. In comparison to the results in Table 8.6, only for the 10–shot experiments, the pure optimization-based methods turn out to be 25 % faster for single inference, whereas they turn out to be slower in the other configurations.

The task adaptation time needed for the various algorithms is provided in Table 8.9. The considered state-of-the-art methods require an adaptation time per task that rises considerably as the number of shots increases. On the contrary, relation-based models, thanks to their comparison-based topology, do not require adaptation for new tasks and therefore lead to a null adaptation time. This results in a great advantage for relational topologies

over traditional optimization-based topologies.

Table 8.9: Adaptation time per new task by algorithm and number of shots. The values, computed on Nvidia® Tesla® P4 GPU, are averaged over three repetitions of each experiment for 10,000 tasks.

Avg. Adaptation Time [ms]	**Reptile**	**MAML 2nd**	**MAML+**	**Weighting-Injection Net**[1]	**MAMW**[1]
1–shot	130	135	135	-	-
2–shot	275	286	310	-	-
5–shot	606	660	667	-	-
10–shot	1,261	1,294	1,411	-	-

[1]For MAMW and Weighting-Injection Net, considering only the need to compare the query with the available supports, the adaptation time is null (0 ms).

To test the application limits of the episodic learning approach for radar-based people counting, experiments were also conducted with up to five people per session in the big room B (Section 8.3.6). In this case, five sessions of one minute each per location and number of people were collected and used. Locations A and C were used to generate training tasks, and locations B and D were used for testing tasks. Table 8.10 presents the results obtained on test data for the average of three experiments and 10,000 final tasks. The results for this meta-dataset show similar characteristics to those where an entire room is used exclusively as a test. In general, the two proposed approaches outperform the state of the art regardless of the number of shots. The MAMW proves more stable and performs better in experiments with very few shots (1– and 2–). The Weighting-Injection Net, on the other hand, outperforms MAMW for the 5– and 10– shot approaches. The extension of the counting approach to up to five people and the limitation of radar resolution for close targets in this scenario make generalization more complex. The increased complexity is reflected in the RDIs input instances and features across the different recording locations. For this reason, with a larger number of shots, MAMW performs less well, favoring noise filtering in support samples rather than classification of query instances. Weighting-Injection Net, on the other hand, focuses directly on learning the query class and performs better in this scenario.

In general, although the proposed algorithms outperform the state of the art, they lead to an average accuracy of less than 60 % over the six classes with 10–shots. This unfortunately shows that the purely episodic generalization approach with a few shots is limited to scenarios with a very small number of people. Adaptations to larger and more varied datasets or the use of radar sensors with higher resolution could obviate the current limitations. The weights of the counting model up to 5 people need an in-

memory size of 1,156 KB. This value is slightly larger than the approach of up to 3 people. More information on a single experiment for the adaptation of up to five people is provided in Appendix 8.9.

Table 8.10: Mean classification accuracy achieved by the various algorithms, for people counting (6 classes): B room, B and D locations.

Accuracy [%] B **room test**	**Reptile**	**MAML** 2^{nd}	**MAML**$^+$	**Weighting-Injection Net**	**MAMW**
1–shot	35.05 ± 0.11	36.82 ± 0.13	35.56 ± 0.13	36.00 ± 0.13	44.86 ± 0.16
2–shot	37.12 ± 0.12	41.01 ± 0.13	40.03 ± 0.13	46.60 ± 0.15	48.71 ± 0.13
5–shot	39.25 ± 0.12	44.74 ± 0.13	43.86 ± 0.13	55.69 ± 0.14	50.64 ± 0.14
10–shot	43.19 ± 0.12	48.56 ± 0.13	46.01 ± 0.12	57.83 ± 0.14	56.71 ± 0.14

8.5.2. Active Learning Experiments

Active learning experiments with the Algorithm 2 are intended to demonstrate how meta learning-driven model initialization benefits task fine-tuning. All the experiments have been carried out on the task of radar-based people counting, using 75 % and 25 % of the data collected in the S room as training and testing, respectively. This means that active learning aims to boost the estimation performance in counting people in the entire small room, given all the locations in which the RDIs were collected. Since all the in-room locations are considered at once, the adaptation in this case is more complex than during episodic training. The uncertainty-based experiments used priority scores S_p defined in Equations (8.13), (8.14) and (8.15). As initialization, the parameters θ obtained after the 1–shot episodic learning of Weighting-Injection Net and MAMW on the remaining two environments (M and B) have been used. As D_p grows larger, the experiments are limited to a maximum of five supports per class. The selected number of epochs for the active learning training is 6,000. For each epoch, 4 queries (J) are to be sampled, with A of them labeled using the uncertainty-based approach. Table 8.11 compares the average results from three experiments for each defined S_p score to the random initialization of θ. As can be seen from the table, the results for initialization based on MAMW and Weighting-Injection Net vary very little as the chosen priority score differs. Such initialization, however, leads to a great performance gap compared to the random one, which also features training instability over repetitions. The Weighting-Injection Net also seems to achieve slightly better performance than the MAMW. This is most likely related to the large availability of labeled data, which for a test room setup, makes this method more performant than MAMW (Section 8.5.1). In the case of random initialization, however, the model succeeds in learning almost exclusively when entropy S_e is used as the scoring function. This may be due to the entropy formulation itself, which results in a

more balanced query selection by taking into account the distribution over all classes for the score computation.

Table 8.11: Accuracy on people counting (4 classes), obtained through pool-based sampling active learning. All the S room data have been used for the adaptation. The results are averaged over three experiment repetitions of 6,000 iterations each. The initialization consists of meta-learned weights for the M and B rooms.

Accuracy [%] Small Room S	**Weighting-Injection Net**	**MAMW**	**Random Init.**
S_{LC}	63.09	60.81	31.14
S_{MS}	63.41	59.98	26.46
S_E	63.62	61.54	43.44

The accuracy learning curve for the entropy-based experiments is depicted in Figure 8.16. Adaptation starting with Weighting-Injection Net and MAMW weights exhibits similar accuracy profiles as training epochs progress. Random initialization, on the other hand, not only leads to lower-performing learning but also to instability and experiment failure, collapsing to a 25 % accuracy over the four classes. In this case, the algorithm encounters difficulties with only a few learning data at a time to generalize to all locations. Fluctuations in accuracy curves are due to adaptation to new labeled data sampled from different S room locations, which normally display different features. This behavior can be observed in the t-SNE representations of the data in Section 8.3.6.

8.6. Conclusion

This paper features how meta learning and active learning can be effectively employed for radar-based people counting using real-world data. For such a use case, multiple meta-datasets are generated based on different combinations of rooms and radar orientations. Episodic learning for few-shot adaptation is carried out through a comparative approach. The model learns task-wise to map features of query examples to representative support instances belonging to the same class. In this way, the belonging class of a radar instance is predicted by comparing it with representative support examples of classes zero to three people. With respect to the traditional weighting network, an injection module increases the input data dimensionality before the comparison step. This process facilitates the comparison of query and support features, reducing episodic task overfitting and aiding generalization. The overall topology with an injection module is called the Weighting-Injection Net.

An episodic adaptation algorithm called model-agnostic meta-weighting

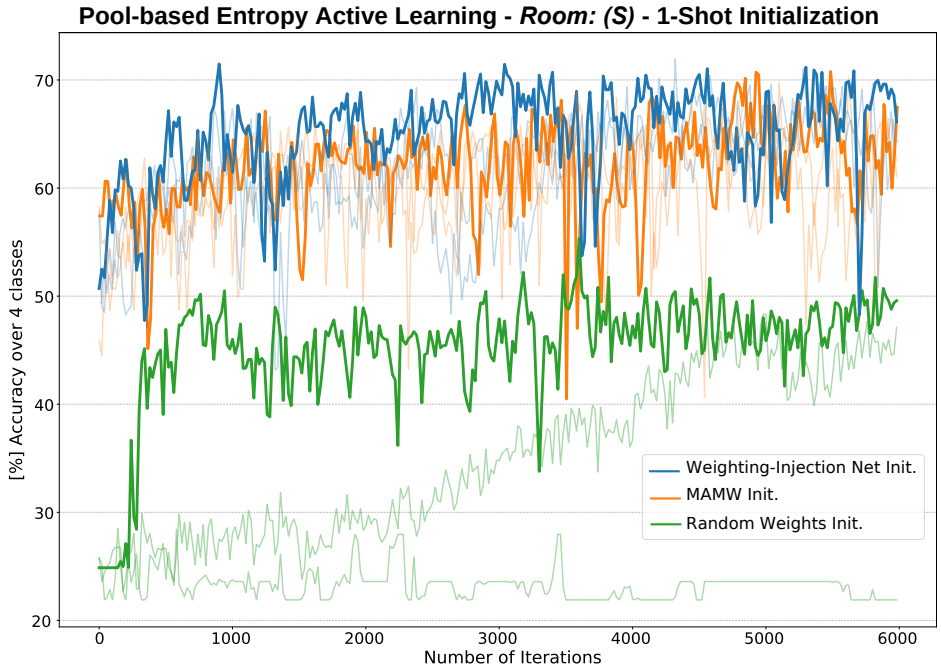


Figure 8.16: Entropy pool-based active learning accuracy across epochs. The thicker lines highlight the best experiments by type of initialization. Accuracy values are averaged per trial every 20 epochs. Random initialization (green) experiments are more unstable and collapse to 25 % random learning on 4 classes.

is then presented for specific adaptations to very few-shot per task. This two-step training algorithm combines the weighting network topology and the optimization-based meta learning approach to enhance the feature extraction capabilities of the model. The approach features an inner step task adaptation that compares support instances with a noisy version of themselves, leading to more stable generalization training, especially in the 1–shot training. Finally, a pool-based active learning approach designed specifically for relation-based methods is presented. Using only the available samples with the highest prediction uncertainty, this algorithm seeks to minimize the number of examples needed for learning.

The presented meta learning achieves cutting-edge accuracy in people counting while also yielding other performance advantages. The relation-based topology grants no training time for adaptation at new radar test locations. Furthermore, the availability of multiple support examples per class allows for more robust averaged query estimation. Both the presented algorithms are up to 15 % more accurate than the state-of-the-art for 1– and 10–shot. They are also found to be up to 50 % faster for computing single-sample inference when the model is tested on a new task. The active

learning algorithm performs better and is more stable when the initialization is set to the episodically learned weights rather than at random. Nonrandom initialization improves radar adaptation accuracy by 30 % on test room radar instances.

Despite the great benefits shown, the work presented is only tested offline on previously collected data. In the future, it will be important to test such a system in a real-time setting. The monitoring approach with more than three people leads to accuracy performance which may be insufficient in several practical contexts. Future work will focus on using relation-based topologies and sensor fusion to counter the current limitations. The use of an unconventional injection module for the relational networks could bring additional benefits for feature representation in episodic learning. In-depth studies will therefore be conducted on the possible applications and limitations of such a module. Research on the injection module will also be carried out in the field of the interpretability of neural networks and training complexity. Also, further active learning and uncertainty sampling strategies that focus on episodic learning with relation-based approaches will be investigated.

8.7. Acknowledgments & Declarations

- **Funding** This work has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No. 876925 (ANDANTE). The JU receives support from the European Union's Horizon 2020 research and innovation programme and France, Belgium, Germany, Netherlands, Portugal, Spain, Switzerland.

- **Code Availability.** The codes for the meta learning algorithms on Omniglot are available at:

  `https://github.com/GiancoMauro/TF-Meta-Learning/`

- **Availability of Data and Materials.** The data are not publicly available due to internal company board policy.

- **Conflicts of Interests.** The authors declare no conflicts of interest.

- **Consent to Participate.** Informed consent was obtained from all subjects involved in the study.

- **Ethics Approval.** The work does not include any personal data relatable to identifiable living persons. The methods and results presented in this paper represent a general radar-based solution for nonuser-specific people counting. No personal information or photos have been obtained from participants.

- **Open Access.** Funding for open acces charge: Universidad de Granada / CBUA (Consorcio de Bibliotecas Universitarias de Andalucía).

8.8. Appendix A. Experiments on Public Dataset

This section presents the results obtained with the Weighting-Injection Net (Section 8.4.1.1) and MAMW (Section 8.4.1.2) on Omniglot [53] public dataset.

8.8.1. Omniglot Dataset

Omniglot [53], is a dataset specifically created for few-shot learning. That dataset contains hand-written instances of as many as 1,623 characters taken from 50 alphabets. Each character was drawn by different people a total of 20 times each. The meta-dataset, divided between $\mathcal{D}^{m-train}$ and $\mathcal{D}^{m-test}$, as defined in the Ominglot repository [2], was used for training and testing the Weighting-Injection Net and MAMW.

8.8.2. Experiments on Omniglot

On Omniglot, the experiments have been performed with 1–shot for 2– and 5–way and 5–shot for 5– and 10–way. The topology used for these experiments is the same as for radar-based people counting (Figure 8.1 and Figure 8.11), but it has been adapted to the larger input size. Each handwritten sample has a resolution of 105×105 pixels. The chosen embedding size g and feature size for the injection module have been 32 and 22, respectively. The configuration of the layers is presented in Table 8.12. In this case, task classification is also accomplished by minimizing categorical crossentropy with Adam as the optimization method, with β_1 and β_2 set to 0 and 0.5, respectively. The episodic learning rate used for the Weighting-Injection Net and the outer step learning rate used for MAMW experiments have been set to $3e-4$ For the MAMW, an inner step learning rate of $5e-5$ has been utilized. Regardless of the number of shots, one query sample q_j per class per episode is used for tasks sampled on $p(\mathcal{T}_r)$ and defined on $\mathcal{D}^{m-train}$. The generalization is then tested episode-wise on 10 test instances per class, on tasks $\mathcal{T}$ sampled from $p(\mathcal{T}_s)$ in $\mathcal{D}^{m-test}$, and $p(\mathcal{T}_r)$. All the experiments have been performed on 22,000 episodes. The built models have then been tested for 10,000 final tests on tasks sampled from $p(\mathcal{T}_s)$ in $\mathcal{D}^{m-test}$.

8.8.3. Results and State-of-the-Art Comparison Omniglot

Also on Omniglot, to assess the generalization performance, box plots have been generated based on the average accuracy obtained for sets of 2,200 episodes. As an example, the trend obtained for the 5–shot, 5–way Weighting-Injection Net experiment is shown in Figure 8.17. As the episodes progress, training on Omniglot sees a sharper increase in generalization in

[2] Available at `https://github.com/brendenlake/omniglot/`

Table 8.12: Network Layers Configuration - Omniglot.

Module	**Type**	**Filter Shape**[1]	**Output Shape**
Injection	Conv2D	2×2×1×64	j×104×104×64
	MaxPool	2×2	j×52×52×64
	Conv2D	3×3×64×64	j×50×50×64
	MaxPool	2×2	j×25×25×64
	Conv2D	2×2×64×64	j×24×24×64
	Conv2D	3×3×64×64	j×22×22×g
Comparison	Conv2D[2]	3×1×2g×g	jc×23×24×g
	MaxPool	3×3	jc×7×8×g
	Conv2D	3×3×g×g	jc×5×6×g
	AvgPool	1×1	jc×g
Weighting	Dense	ng×64	j×64
	Dense	1×n	j×n

Table 8.13: The mean classification accuracy achieved by the various selected algorithms for experiments on Omniglot.

Accuracy Omniglot Eval. [%]	**Reptile**	**MAML 2nd**	**MAML+**	**Weighting-Injection Net**	**MAMW**
1–shot 2–way	69.21 ± 0.30	74.18 ± 0.34	80.30 ± 0.32	76.65 ± 0.32	81.99 ± 0.31
1–shot 5–way	41.14 ± 0.10	55.76 ± 0.23	59.36 ± 0.23	71.46 ± 0.23	72.19 ± 0.23
5–shot 5–way	52.59 ± 0.20	84.99 ± 0.14	77.50 ± 0.18	85.76 ± 0.15	85.02 ± 0.16
5–shot 10–way	36.72 ± 0.15	78.60 ± 0.12	77.61 ± 0.13	79.11 ± 0.12	81.23 ± 0.12

the early stages than radar-based people counting. This is most likely caused by the greater variety of classes among the handwritten characters, whose features take longer to be extracted by the relational network through the comparison mechanism.

The accuracy values achieved with Weighting-Injection Net and MAMW are listed for the various experiments in Table 8.13, in comparison with state-of-the-art methods. For the state-of-the-art algorithms, a TensorFlow™ implementation and the same testing pipeline as for the people counting comparison have been adopted. The accuracy of the tasks is not calculated on a single query sample per class, as in Reptile [32], but on ten test instances per class in a step following the learning step. This allows a more fair comparison with relational algorithms, where the query example is not used in a step subsequent to the support ones. In addition, no data augmentation or scaling is performed on single inputs, in contrast to the MAML methods presented in [31, 52]. For the state-of-the-art methods, the same CNN topology and configurations presented in Section 8.5.1.1 for radar-based people counting

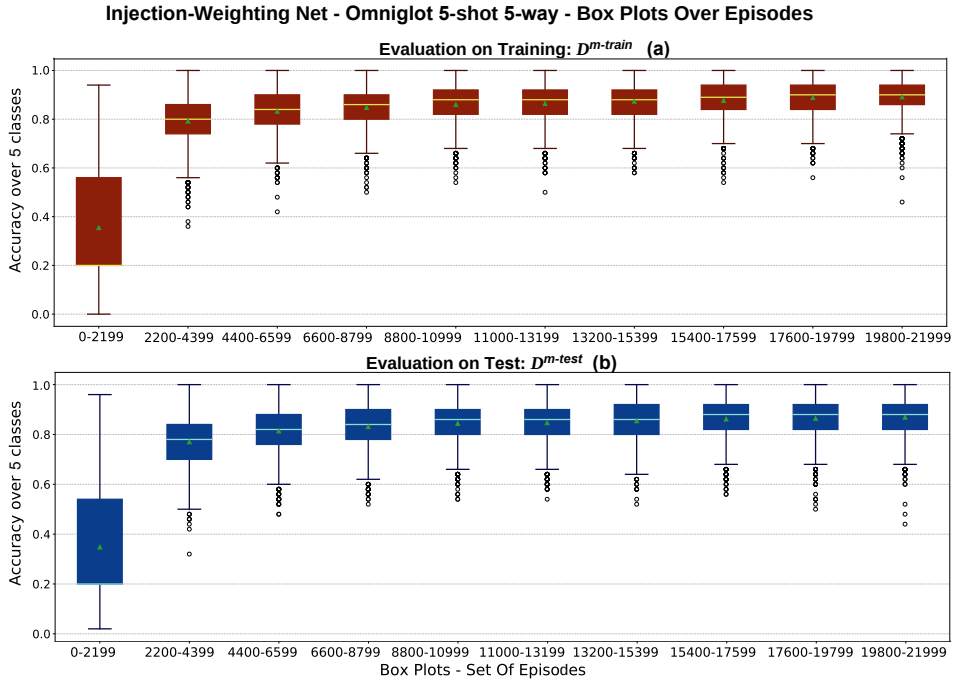


Figure 8.17: Accuracy statistics box plots vs. episodes for a Weighting-Injection Net 5–shot 5–way experiment on Omniglot. The red box plots are constructed on validation tasks sampled from $\mathcal{D}^{m-train}$ (**a**), whereas the blue ones are constructed on test tasks sampled from $\mathcal{D}^{m-test}$ (**b**). The median and mean values are represented by a horizontal line and a green triangle in each box plot.

have been used on Omniglot.

All experiments have been performed on an Nvidia® Tesla® P4 GPU and CUDA® Toolkit v11.1.0 for parallel computing.

Similarly to what has been observed in Section 8.5.1 for the radar-based people counting dataset, the MAMW seems to perform better than the Weighting-Injection Net in the 1–shot and 10–way scenarios (Table 8.13). For the 5–shot 5–way experiment, the two relation-based algorithms achieved similar accuracy, which is comparable to MAML 2$^{\text{nd}}$. This may be inherent in the fact that for Omniglot, unlike radar data, there is no intrinsic background noise in the input instances. Consequently, the introduction of noise in the comparison between supports in MAMW does not promote generalization learning when many shots are fed to the network. Conversely, MAMW inner step may divert attention away from the learning goal of single tasks. Even for Omniglot, using the injection module seems to help generalization learning by making it easier to compare support features and queries in a higher dimension. Regardless of the number of ways and shots, the

Weighting-Injection Net and MAMW outperform the other state-of-the-art in most of the Omniglot experiments. The presented methods, with about 30 % fewer parameters, also perform significantly better in single-shot approaches than optimization-based methods. In the 1–shot 5–way experiment, MAMW leads to an average accuracy about 18 % higher than MAML$^+$.

8.9. Appendix B. More People Count Details

This section analyses a single episodic meta learning experiment for radar-based indoor people counting up to five people.

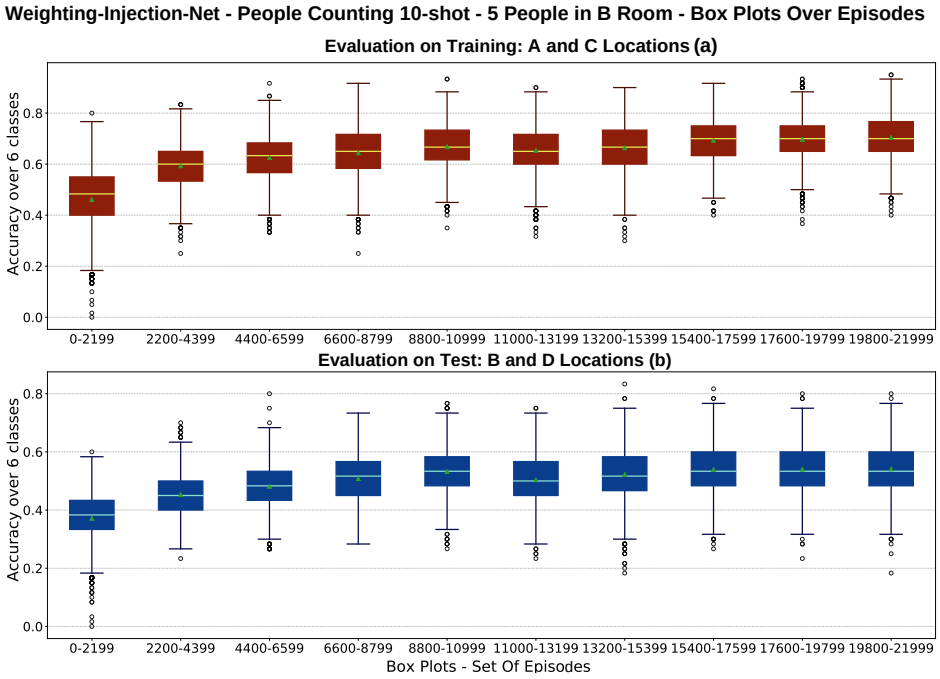


Figure 8.18: Accuracy statistics box plots vs. episodes for a Weighting-Injection Net 10–shot 6–way experiment on radar-based people counting (B room). The red box plots are constructed on validation tasks (**a**), whereas the blue box plots are constructed on test tasks (**b**). The median and mean values are represented by a horizontal line and a green triangle in each box plot.

8.9.1. Single Experiment People Counting Analysis up to Five Individuals.

The outcomes of the episodic adaptation on the five people meta-dataset of Section 8.5.1.1 can be analyzed at the level of the individual experiment.

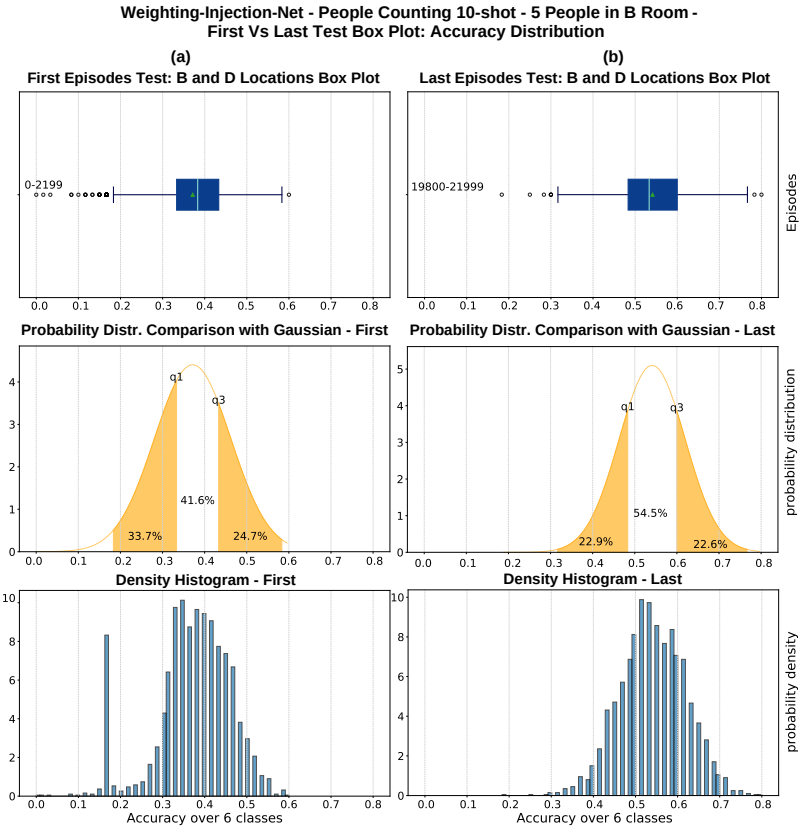


Figure 8.19: Weighting-Injection Net 10–shot 6–way, first (**a**) and last (**b**) box plot underlying distributions, generated on people counting test tasks. The q1 and q2 values on the Gaussians indicate the first and third quartiles, respectively. The probability density histograms show the actual non-Gaussian nature of the distribution. The accuracy probability density for the last box plot (**b**) has a mean value shifted towards higher accuracy as a result of the generalization learning.

Every experiment consists of 22,000 episodes of meta-training in the room B (Figure 8.5). Training and validation are performed on tasks sampled from locations A and C in the room, while testing is done on tasks sampled from locations B and D. The experiment is a 6–way, since zero individuals in the room is also considered a class. Figures 8.18, 8.19 and 8.20, show different statistical insights of a 10–shot Weighting-Injection Net experiment. Figure 8.18 displays the trend of box plots built on accuracy as episodes increase. Compared to the training up to three people (Figure 8.12), the adaptation up to five people shows a less pronounced trend of improvement. In this case, the test fails to generalize better from 15,000 episodes onward, reaching a saturation of accuracy around 55 %. Figure 8.19 reveals the den-

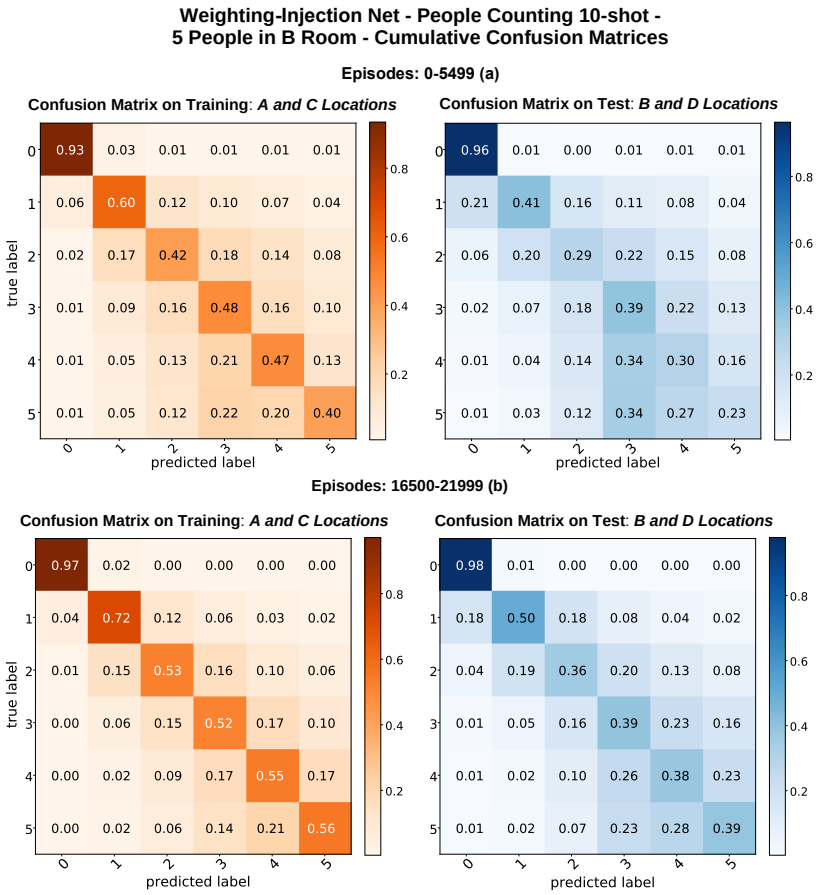


Figure 8.20: Cumulative confusion matrices for Weighting-Injection Net 10–shot 6–way people counting experiment. Confusion matrices are obtained on the first (**a**) and last (**b**) 5,550 meta-iterations in the validation phase for both training and test sampled tasks.

sity histograms underlying the first and last box plots constructed on the test in episodic learning. In comparison to the adaptation of up to three people Figure 8.13, no marked reduction in whiskers or negative skew in the last histogram is noticeable. Yet, there is an increase in average accuracy from 37 % to 55 % (18 % improvement in generalization). A very interesting analysis can be done by analyzing the accuracy on individual classes, thus by generating the cumulative confusion matrices shown in Figure 8.20. As in the confusion matrices generated for the 4-way approach (Figures 8.14 and 8.15), the model easily succeeds in classifying the absence of people in the environment, reaching a solid 98 % class accuracy in the test at the end of episodic learning. Further, as the episodes progress, the generalization approach yields higher accuracy in counting more than one person. Moreover,

most of the miss-classifications lie around the main diagonal of the confusion matrix, which represents the ± 1 of accuracy. This means that most of the classification errors tend to under- or overestimate the number of people in the room by only one unit.

References

[1] Elisabetta Moisello, Piero Malcovati, and Edoardo Bonizzoni. Thermal sensors for contactless temperature measurements, occupancy detection, and automatic operation of appliances during the covid-19 pandemic: A review. *Micromachines*, 12(2):148, 2021.

[2] ASF Rahman, SB Yaakob, ARA Razak, and RA Ramlee. Post covid-19 implementation of a bidirectional counter with reduced complexity for people counting application. In *Journal of Physics: Conference Series*, volume 1878, page 012040. IOP Publishing, 2021.

[3] Ahmad Taha, Jan Krabicka, Ruiheng Wu, Peter Kyberd, and Neil Adams. Design of an occupancy monitoring unit: a thermal imaging based people counting solution for socio-technical energy saving systems in hospitals. In *2019 11th Computer Science and Electronic Engineering (CEEC)*, pages 1–6. IEEE, 2019.

[4] Ya-Li Hou and Grantham KH Pang. People counting and human detection in a challenging situation. *IEEE transactions on systems, man, and cybernetics-part a: systems and humans*, 41(1):24–33, 2010.

[5] G Thomas Prathiba and YP Dhas. Literature survey for people counting and human detection. *IOSR Journal of Engineering (IOSRJEN)*, 3(1):05–10, 2013.

[6] Chakravartula Raghavachari, V Aparna, S Chithira, and Vidhya Balasubramanian. A comparative study of vision based human detection techniques in people counting applications. *Procedia Computer Science*, 58:461–469, 2015.

[7] Michal Stec, Viktor Herrmann, and Benno Stabernack. Using time-of-flight sensors for people counting applications. In *2019 Conference on Design and Architectures for Signal and Image Processing (DASIP)*, pages 59–64. IEEE, 2019.

[8] Antonio Brunetti, Domenico Buongiorno, Gianpaolo Francesco Trotta, and Vitoantonio Bevilacqua. Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing*, 300:17–33, 2018.

[9] Naveed Ilyas, Ahsan Shahzad, and Kiseon Kim. Convolutional-neural network-based image crowd counting: review, categorization, analysis, and performance evaluation. *Sensors*, 20(1):43, 2019.

[10] Faisal Abdullah, Yazeed Yasin Ghadi, Munkhjargal Gochoo, Ahmad Jalal, and Kibum Kim. Multi-person tracking and crowd behavior detection via particles gradient motion descriptor and improved entropy classifier. *Entropy*, 23(5):628, 2021.

[11] Saleh Basalamah, Sultan Daud Khan, and Habib Ullah. Scale driven convolutional neural network model for people counting and localization in crowd scenes. *IEEE Access*, 7:71576–71584, 2019.

[12] Marina Ivasic-Kos, Mate Kristo, and Miran Pobar. Person detection in thermal videos using yolo. In *Proceedings of SAI Intelligent Systems Conference*, pages 254–267. Springer, 2019.

[13] Sylvia T Kouyoumdjieva, Peter Danielis, and Gunnar Karlsson. Survey of non-image-based approaches for counting people. *IEEE Communications Surveys & Tutorials*, 22(2):1305–1336, 2019.

[14] Atika Gupta, Sudhanshu Maurya, Nidhi Mehra, and Divya Kapil. Covid-19: Employee fever detection with thermal camera integrated with attendance management system. In *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 355–361. IEEE, 2021.

[15] Asad Lesani, Ehsan Nateghinia, and Luis F Miranda-Moreno. Development and evaluation of a real-time pedestrian counting system for high-volume conditions based on 2d lidar. *Transportation research part C: emerging technologies*, 114:20–35, 2020.

[16] Andrei Günter, Stephan Böker, Matthias König, and Martin Hoffmann. Privacy-preserving people detection enabled by solid state lidar. In *2020 16th International Conference on Intelligent Environments (IE)*, pages 1–4. IEEE, 2020.

[17] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7356–7365, 2018.

[18] Fei Wang, Sanping Zhou, Stanislav Panev, Jinsong Han, and Dong Huang. Person-in-wifi: Fine-grained person perception using wifi. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5452–5461, 2019.

[19] Johannes Rimmelspacher, Radu Ciocoveanu, Giovanni Steffan, Matteo Bassi, and Vadim Issakov. Low power low phase noise 60 ghz multi-channel transceiver in 28 nm cmos for radar applications. In *2020 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*, pages 19–22. IEEE, 2020.

[20] Radu Ciocoveanu and Vadim Issakov. Low-power 60ghz receiver with an integrated analog baseband for fmcw radar applications in 28nm cmos technology. In *2021 IEEE 20th Topical Meeting on Silicon Monolithic Integrated Circuits in RF Systems (SiRF)*, pages 4–6. IEEE, 2021.

[21] Saverio Trotta, Dave Weber, Reinhard W Jungmaier, Ashutosh Baheti, Jaime Lien, Dennis Noppeney, Maryam Tabesh, Christoph Rumpler, Michael Aichner, Siegfried Albel, et al. Soli: a tiny device for a new human machine interface. In *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, volume 64, pages 42–44. IEEE, 2021.

[22] Avik Santra and Souvik Hazra. *Deep learning applications of short-range radars*. Artech House, 2020.

[23] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[24] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR)*, pages 1–14. San Diego, CA, USA, 2015.

[25] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[26] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

[27] Xiaoxu Li, Zhuo Sun, Jing-Hao Xue, and Zhanyu Ma. A concise review of recent few-shot meta-learning methods. *Neurocomputing*, 456:463–468, 2021.

[28] Abdullatif Köksal, Timo Schick, and Hinrich Schütze. Meal: Stable and active learning for few-shot prompting. *arXiv preprint arXiv:2211.08358*, 2022.

[29] Joaquin Vanschoren. Meta-learning. *Automated machine learning: methods, systems, challenges*, pages 35–61, 2019.

[30] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169, 2021.

[31] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[32] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[33] Rabindra Khadka, Debesh Jha, Steven Hicks, Vajira Thambawita, Michael A Riegler, Sharib Ali, and Pål Halvorsen. Meta-learning with implicit gradients in a few-shot setting for medical image segmentation. *Computers in Biology and Medicine*, page 105227, 2022.

[34] Gianfranco Mauro, Mateusz Chmurski, Lorenzo Servadei, MP Cuellar, and Diego P Morales-Santos. Few-shot user-definable radar-based hand gesture recognition at the edge. *IEEE Access*, 2022.

[35] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

[36] Xianglong Zeng, Chaoyang Wu, and Wen-Bin Ye. User-definable dynamic hand gesture recognition based on doppler radar and few-shot learning. *IEEE Sensors Journal*, 21(20):23224–23233, 2021.

[37] Punit Kumar and Atul Gupta. Active learning query strategies for classification, regression, and clustering: a survey. *Journal of Computer Science and Technology*, 35:913–945, 2020.

[38] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM Computing Surveys (CSUR)*, 54(9):1–40, 2021.

[39] Lucas Massa, Adriano Barbosa, Krerley Oliveira, and Thales Vieira. Lrcn-retailnet: A recurrent neural network architecture for accurate people counting. *Multimedia Tools and Applications*, 80(4):5517–5537, 2021.

[40] Andres Gomez, Francesco Conti, and Luca Benini. Thermal image-based cnn's for ultra-low power people recognition. In *Proceedings of the 15th ACM International Conference on Computing Frontiers*, pages 326–331, 2018.

[41] Sanaz Kianoush, Stefano Savazzi, Vittorio Rampa, and Monica Nicoli. People counting by dense wifi mimo networks: Channel features and machine learning algorithms. *Sensors*, 19(16):3450, 2019.

[42] Runhan Bao and Zhaocheng Yang. Cnn-based regional people counting algorithm exploiting multi-scale range-time maps with an ir-uwb radar. *IEEE Sensors Journal*, 21(12):13704–13713, 2021.

[43] Michael Stephan, Souvik Hazra, Avik Santra, Robert Weigel, and Georg Fischer. People counting solution using an fmcw radar with knowledge distillation from camera data. In *2021 IEEE Sensors*, pages 1–4. IEEE, 2021.

[44] Jennifer Vandoni, Emanuel Aldea, and Sylvie Le Hégarat-Mascle. Active learning for high-density crowd count regression. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2017.

[45] Zhen Zhao, Miaojing Shi, Xiaoxiao Zhao, and Li Li. Active crowd counting with limited supervision. In *European Conference on Computer Vision*, pages 565–581. Springer, 2020.

[46] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 589–597, 2016.

[47] Mahesh Kumar Krishna Reddy, Mohammad Hossain, Mrigank Rochan, and Yang Wang. Few-shot scene adaptive crowd counting using meta-learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2814–2823, 2020.

[48] Changtong Zan, Baodi Liu, Weili Guan, Kai Zhang, and Weifeng Liu. Learn from object counting: Crowd counting with meta-learning. *IET Image Processing*, 15(14):3543–3550, 2021.

[49] Xiaoyu Hou, Jihui Xu, Jinming Wu, and Huaiyu Xu. Cross domain adaptation of crowd counting with model-agnostic meta-learning. *Applied Sciences*, 11(24):12037, 2021.

[50] Huawei Hou, Suzhi Bi, Lili Zheng, Xiaohui Lin, Yuan Wu, and Zhi Quan. Dasecount: Domain-agnostic sample-efficient wireless indoor crowd counting via few-shot learning. *IEEE Internet of Things Journal*, 2022.

[51] Yong Zhang, Yang Chen, Yujie Wang, Qingqing Liu, and Andong Cheng. Csi-based human activity recognition with graph few-shot learning. *IEEE Internet of Things Journal*, 9(6):4139–4151, 2021.

[52] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. In *7th International Conference on Learning Representations (ICLR)*, page Poster. New Orleans, LA, USA, 2019.

[53] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Sotto l'azzurro fitto
del cielo qualche uccello di mare se ne va;
né sosta mai: perché tutte le immagini portano scritto:
"più in là!"

Maestrale
Ossi di Seppia
Eugenio Montale