Luís Fernando do Vale Santana

Exploração de Radar para Reconhecimento de Gestos

Exploring Radar Sensing for Gesture Recognition

Universidade de Aveiro
2021

Luís Fernando do Vale Santana

Exploração de Radar para Reconhecimento de Gestos

Exploring Radar Sensing for Gesture Recognition

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Doutor Samuel de Sousa Silva, Professor Auxiliar do da Universidade de Aveiro, e da Doutora Ana Patrícia Oliveira Ferreira da Rocha, Investigadora do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

This work was developed in the scope of the research project APH-ALARM: Comprehensive safety solution for people with Aphasia - AAL/0006/2019 - , funded by national funds (OE), through FCT/MCTES

o júri / the jury

presidente / president

Professor Doutor Paulo Miguel de Jesus Dias
Professor Auxiliar, Universidade de Aveiro

vogais / examiners committee

Doutor Francisco José Curado Mendes Teixeira
Consultor Para o Ensino Superior, Secção Economia Verde, Inovação e Tecnologia - Comissão Económica Para Africa das Nações Unidas

Professor Doutor Samuel de Sousa Silva
Professor Auxiliar em Regime Laboral, Universidade de Aveiro

agradecimentos / acknowledgements

Queria agradecer a minha familia pelo apoio e incentivo ao longo dos anos. Agradeço também ao Bruno e ao Espincho por todas as palavras encorajadoras e pelos bons momentos que limpavam pesos dos meus ombros. Um obrigado também ao Nimpu pelas conversas sobre o mundo, a vida e também carros. Um grande abraço também aos "Esgaça" por todas as saidas, todos as noitadas a jogar e todos os jantares. Por fim agradeço à Mariana por aturar todo o meu mau humor e pessimismo, e por retribuir com carinho e afeto. Muito Obrigado.

Palavras Chave

Radar, afasia, gestos, comunicação, reconhecimento gestual, casa inteligente, *transfer learning*, sensores não intrusivos

Abstract

Os problemas de comunicação têm um efeito nocivo nas vidas das pessoas como isolamento, depressão e perda de independência. Ao longo dos anos, várias abordagens para atenuar estes problemas foram propostas, sendo que a maioria tem desvantagens. Falta de versatilidade, soluções intrusivas ou a necessidade de andar com um dispositivo são alguns dos problemas destas soluções.

O uso de radares tem visto um aumento nos últimos anos, chegando até áreas variadas como o setor de saúde ou automóvel. Este tipo de solução é não intrusiva, não é sensível a mudanças das condições ambientais como luz e não invade a privacidade do utilizador como o uso de câmaras.

Nesta dissertação e no âmbito do projeto APH-ALARM, testou-se um radar no contexto do reconhecimento de gestos para apoio à comunicação no cenário do quarto. Neste cenário, o utilizador é alguém com problemas de comunicação, que se encontra deitado na sua cama e precisa de comunicar com um familiar dentro ou fora de casa. O uso de gestos permite ao utilizador ter algum apoio durante a comunicação e ajuda o mesmo a expressar as suas necessidades.

Para reconhecer os gestos feitos pelo utilizador, é necessário capturar o movimento humano. Para demonstrar as capacidades da tecnologia para este contexto, foi implementada uma prova de conceito de um sistema que captura os dados do radar e de seguida os filtra, converte-os em imagens e usa as mesmas como entrada de um modelo para classificação de gestos.

Para avaliar a solução proposta, foram recolhidos dados de quatro pessoas enquanto realizavam dez repetições de cinco gestos diferentes com um dos braços. Uma solução independente do utilizador mostrou ser um caso mais desafiante quando comparada com uma solução dependente do utilizador, em que todos os datasets excepto um tem um acerto médio superior a 70% em que a maioria deles supera os 90%.

Keywords

Radar, aphasia, gesture, communication, gesture recognition, smart home, transfer learning, non-intrusive sensors

Abstract

Communication disorders have a notable negative impact on people's lives, leading to isolation, depression and loss of independence. Over the years, many different approaches to attenuate these problems were proposed, although most come with noticeable drawbacks. Lack of versatility, intrusive solutions or the need to carry a device around are some of the problems that these solutions encounter.

Radars have seen an increase in use over the past few years and even spreading to different areas such as the automotive and health sectors. This technology is non-intrusive, not sensitive to changes in environmental conditions such as lighting, and does not intrude on the user's privacy unlike cameras.

In this dissertation and in the scope of the APH-ALARM project, the author tests the radar in a gesture recognition context to support communication in the bedroom scenario. In this scenario, the user is someone with communication problems, lying in their bed trying to communicate with a family member inside or outside the house. The use of gestures allows the user to have assistance communicating and helps express their wants or needs. To recognize the gestures executed by the user, it is necessary to capture the movement. To demonstrate the capabilities of the technology, a proof of concept system was implemented, which captures the data, filters and transforms it into images used as input for a gesture classification model.

To evaluate the solution, we recorded ten repetitions of five arm gestures executed by four people. A subject independent solution proved to be more challenging when compared to a subject dependent solution, where all datasets but one achieved a median accuracy above 70% with most going over 90%.

Contents

List of Figures

List of Tables

Glossary

radar	Radio Detection And Range
FMCW	Frequency Modulated Continuous Wave
NIDCD	National Institute on Deafness and Other Communication Disorders
AAC	Augmentative and Alternative Communication
PwA	Person with Aphasia
TTS	Text-To-Speech
IOS	Iphone Operating System
MacOS	Mac Operating System
UCLA	University of California, Los Angeles
PCA	Principal Component Analysis
KNN	K-Nearest Neighbors
DTW	Dynamic Time Warping
CNN	Convolutional Neural Network
HMM	Hidden Markov Model
SNR	Signal to Noise Ratio
IF	Intermediate Frequency
ADC	Analogue-to-Digital Converter
DSP	Digital Signal Processing
FFT	Fast Fourier Transform
PNN	Probabilistic Neural Network
TLV	Tag Length and Values
ASHA	American Speech-Language-Hearing Association

CHAPTER 1

Introduction

1.1 Context and Motivation

Communication is the act of exchanging ideas, emotions and thoughts to one another [1]. As human beings are inherently social creatures, communication is one of the most paramount skills a person can have. In our everyday lives, we communicate with others, be it verbally or non-verbally, in a personal or professional way. It is how we interact and make connections with each other. And that is why communication difficulties have such a negative effect on people. The impacts of those problems include the inability to live in society and do everyday chores which can lead to lack of self-esteem, isolation, mental health problems, a loss of independence, and a sense of insecurity.

According to the National Health Interview Survey from 2012, around 8% of children in the U.S. were diagnosed with a communication disorder [2]. Plus, 7.5 million Americans have trouble using their voice, as reported by the National Institute on Deafness and Other Communication Disorders [2]. The American Speech-Language-Hearing Association divides communication disorders into four types [3]:

- Speech disorder: impairs the user's articulation and fluency;
- Language disorder: impairs the user's comprehension or use of spoken or written symbols;
- Hearing disorder: impairs the user's auditory sensitivity;
- Central auditory processing disorders: impairs the user's ability to organize, transform, store of information contained in audible signals.

One of the many possible communication disorders is known as Aphasia. Aphasia impairs the person's capability of understanding others and expressing themselves [4]. It can also lead to difficulties in reading and writing. This is an acquired disorder most commonly caused by strokes. According to the National Aphasia Association, about one third of stroke survivors

develop the illness, with around two million Americans and 250,000 people in Great Britain being affected by it [5]. Someone affected by the disease can still lead a regular life, although complex social interactions are affected by the condition.

In the past years, many different processes or approaches that augment, complement or replace speech were created to support people with communication disorders [6]. They are usually referred to as Augmentative and Alternative Communication (AAC), and, by providing these features, all have the same goal, to help people regain independence, self-esteem and, at the same time, feel more secure. AAC ranges from the use of the user's body to convey non-verbal messages through facial expressions or voluntary motor movements (e.g., sign language) to books or physical display boards with images and phrases or the use of electronic devices together with software applications, like smart devices (e.g., tablet, smartphone) with image board apps. Other AAC approaches use different devices or sensors to interact with the user [6]. Mechanical switches or keyboards integrated into other devices, such as computers, are very prominent in this area. As mentioned previously, the use of touchscreen on tablets or smartphones is another possibility. Another approach is the use of cameras to track the users head or eyes. It is also possible to translate the brains electrical impulses into a message. In the vast majority of the cases, the output is transformed into symbols or digitized speech.

1.2 Limitations and Challenges of AAC

Although the existing AAC approaches provide acceptable solutions for most cases, there are still some limitations present in them. In most cases, these devices require the user to carry equipment with them. In some instances of AAC, the use of RGB cameras is present. This use makes the tools very vulnerable to environmental conditions such as light or dust, which can hurt their performance. Besides this vulnerability, the use of cameras comes with privacy issues that need addressing. Picture boards have a limited number of pictures on them, which can be detrimental in some cases.

Other types of technologies come with different disadvantages. Brain-computer interfaces decode the users' electrical impulses and translate the signal into a message. This method can be invasive or non-invasive. Invasive BCIs use electrodes implanted in the brains' peripheral nerves. Non-invasive BCIs use flat metal discs placed in the users' scalp. This technique is called an electroencephalogram. Both cases offer disadvantages, one being invasive and the other requiring the user to carry the sensors. The use of keyboards and switches is also present in AAC. Although this approach allows for some versatility, it is not the easiest to use while lying in bed [6].

From these limitations come a few challenges for AAC solutions:

- Being easy to carry/wear, or not requiring the user to carry the device with them at all or wear any sensor;
- Avoid the privacy issues that come with the use of cameras;
- Being independent of environmental conditions such as light;
- Being easy to use at any time of day, including daytime and nighttime;
- Being easy to use when asking for help

1.3 APH-ALARM Project

In this context, the ongoing project APH-ALARM – Comprehensive safety solution for people with Aphasia (AAL/0006/2019) [1], aims at allowing people suffering from aphasia (e.g., after a stroke) to communicate more easily with other people anywhere and anytime. A smartphone application is under development in which the user utilizes pictograms to communicate. In a situation where the user can not use the pictograms, the application can also interpret gestures. The user is also able to choose to send the message to someone. The project also includes a solution for the bedroom scenario, where the user is lying in bed. This solution involves sensors placed on the bed, the user or scattered around the bedroom. In the scope of this project, our main objective is to enhance communication for people with speech difficulties, in the in-bed scenario (i.e., user lying in bed).

1.4 Objectives

The main aim of this work is to explore technologies that can be used to provide people that have communication impairments/disorders with solutions that minimize their communication difficulties, while also addressing some of the problems challenges mentioned above. Regarding the explored technology, the focus is on the Radio Detection And Range (radar) for numerous reasons. It has shown strong potential in the last couple of years for various applications. It also does not require the user to carry an device with them or wear any device or sensor on their body, and is not vulnerable to environmental conditions like cameras. To achieve the defined goal the following sub-objectives were outlined:

- Perform a revision of the literature regarding communication problems to understand and identify the challenges that need addressing, in particular, Aphasia and its effects, as well as existing approaches for AAC and their limitations;
- Carry out a revision of the literature regarding radar and its functioning;
- Gather literature regarding gesture recognition and radar-based gesture recognition;
- Explore the radar technology's capabilities in concerning movement detection and data acquisition, in the context of the considered scenario;
- Design and implement a proof of concept of a system for gesture recognition aiming at supporting communication;
- Evaluate the gesture recognition component of the proof of concept with data collected from different subjects in a real setting, for two different types of solutions: user dependent and user independent.

1.5 Contributions

The work described in this document yielded several contributions, which are:

- A review of the literature regarding AAC, radars and gesture recognition;

[1] http://www.aal-europe.eu/projects/aph-alarm/

- Results of the initial exploration of the radar capabilities, including the most adequate configuration parameters for data acquisition in the specific bedroom scenario;
- A proof of concept of a gesture recognition system showcasing the radar's capabilities;
- A gesture dataset captured from multiple subjects using a radar ;
- A pipeline for gesture recognition, integrated in the proof of concept system, where the input corresponds to data provided by a radar and the output is the name of the gesture;
- Results of gesture recognition evaluation, based on the gesture dataset described above, considering both the user dependent and independent cases.

Some of the contributions have already been accepted for presentation in EAI MobiQuitous 2021, the International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, with an article entitled "Radar-Based Gesture Recognition Towards Supporting Communication in Aphasia: The Bedroom Scenario"(see appendix B):

- Authors: Luís Santana, Ana Patrícia Rocha, Afonso Guimarães, Ilídio C. Oliveira, José Maria Fernandes, Samuel Silva and António Teixeira
- Title: "Radar-Based Gesture Recognition Towards Supporting Communication in Aphasia: The Bedroom Scenario"
- Year: 2021
- Conference: accepted for presentation in EAI MobiQuitous 2021, the International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services

Other contributions include the participation as co-author of two papers on work related to this dissertation, one published [7] and another submitted to the conference PerCom 2021 International Conference on Pervasive Computing and Communications [8].

1.6 Dissertation Structure

This document is divided into six chapters.

Chapter 2 dives into some of the literature deemed appropriate for the work. To start, several topics such as communication disorders, aphasia and AAC. After this, the chapter tackles more technical aspects such as radar technology. These topics provide background regarding influential subjects for the work. After providing the background, the chapter analyses several articles with related work regarding radar and gesture recognition.

The chapter tackles several topics such as communication disorders, AAC, gesture recognition and radar.

Chapter 3 defines and describes some personas and scenarios created with the purpose of helping guide the development of the work and guarantee that it is in line with the challenges identified in the current chapter. The personas and scenarios also helped create the requirements for the system, including both non-functional and functional requirements, which are also listed in Chapter 3.

Chapter 4 describes the proof of concept of a system for gesture recognition and all its development stages. The chapter begins with an overview of the proposed systems' architecture.

Next comes an analysis of the chosen gestures and their description. The last sections of the chapter describe the gesture recognition pipeline, where data captured by the radar are acquired, processed and used as input to a model that classifies the gestures performed by the user.

Chapter 5 describes and analyses the obtained results. The chapter dives into all the preliminary experiments executed during the development of the proof of concept. After describing the first attempts, the results of an evaluation concerning gesture recognition, based on a dataset gathered from multiple subjects, are presented and discussed. The evaluation was carried out for two different types of solutions: user dependent (one model per user) and user independent (model trained with several subjects and tested with a never seen subject).

Chapter 6 wraps up the document with an analysis of the objectives mentioned in the first chapter and whether or not these have been met, a short description of the development stages and the main conclusions that can be drawn from the performed work. The chapter also contains some future work to be done.

Appendix A contains some technical aspects of the radar that have been moved to simplify the text.

Appendix B contains the article entitled "Radar-Based Gesture Recognition Towards Supporting Communication in Aphasia: The Bedroom Scenario".

CHAPTER 2

Background and Related Work

This chapter provides some background and analyses some existing literature regarding communication problems and the methods that try to mitigate their effects. It also discusses the advances in gesture recognition and radar technologies, as well as their advantages.

2.1 Communication Disorders

Communication disorders affect a persons' capacity to communicate with others. These disorders can affect several capabilities of a person such as hearing, speech and language. As mentioned previously, communication problems have a negative effect on people that can lead to isolation and mental health problems. These problems can have several origins, including disorders such as aphasia. Aphasia results from damage to portions of the brain, most commonly on the left side [9]. Its development is slow if caused by a tumour or other progressive neurological diseases, or sudden if it originated from a stroke or head trauma. This disorder affects the person's linguistic capabilities, hindering production and/or comprehension of speech and also reading and writing. As its main cause is strokes, middle-aged and older people are more affected by the disorder, but anyone can acquire it.

As referred to in [10], many people with aphasia mention having strong emotions to deal with in their post-diagnosis life. Emotions such as anger, sadness and sometimes shame. Some of these come from the fact that other people do not usually understand the problem or how to deal with it, making them not want to communicate with the Person with Aphasia (PwA). This factor leads to exclusion and, in some cases, depression. Some aphasics also refer to a loss of independence and/or responsibility, due to the fear of not being able to communicate in an emergency.

2.2 Augmentative and Alternative Communication

Augmentative and Alternative Communication (AAC) comprises approaches that complement, augment or replace speech, with the intent of supporting people with communication

disorders [6]. As mentioned before, a large number of AAC approaches that help people with communication problems already exist. Although these are not specifically for aphasia, they can help mitigate some effects of it. This section will focus on some of the more technology-driven approaches.

These approaches include devices and/or applications that rely on sensing modalities, which can be used in a stand-alone format or in combination with one another [6]. These modalities are associated with different types of activation methods, many of them corresponding to sensors (e.g., keyboard, touchscreen, camera, microphone).

The sensing modalities used in AAC can be divided into the following categories, according to the used method [6]:

- Imaging;
- Mechanical and electromechanical;
- Touch-activated;
- Breath-activated;
- Brain-Computer Interface.

Imaging methods are commonly used to support users with reduced body movement, as they regularly rely on eye gaze or head detection and tracking as the activation method. This method is paired with the use of cameras. A lot of devices of this kind are commercially available, such as the ones provided by Tobii Dynavox [11]. The i-series devices are a speech generating device that provides eye-tracking capability combined with touch access input. They also contain various communication apps that resemble picture boards [11]. Another example of these devices is the WinSlate 12D from Forbes AAC [12]. The device, which provides various methods of activation, such as custom keyguards, switch access, among others [12]. When paired with the Enable Eyes™ tracking module, the device allows the user to simulate all mouse controls with their eyes [13].

Mechanical and electromechanical methods offer direct and indirect access to the selection of input. This means that it can accommodate the needs of both people with limited or regular body movement. Direct selection access revolves around mechanical switches or keyboards, while indirect selection uses a scanning process to access the user's options. The scanning process can either be interval based or controlled by the user [6].

Touch-activated methods are commonly direct access methods, due to the rise of touchscreen technologies and devices. As smartphones have become a prevalent presence in people's everyday lives, several AAC apps for these devices have appeared. These use the touchscreen of the device combined with picture boards to aid the user's communication process [6]. Examples of these apps are the Proloquo2Go from AssistiveWare [14]. This app is available for IOS/MacOS devices only and provides symbol-based AAC and Text-To-Speech (TTS) with a vocabulary of over 10,000 words. Another example is Predictable from Therapy Box [15]. Another TTS AAC app available for both IOS and Android that combines picture boards with TTS capabilities. Besides this, Predictable also allows the user to select other methods of interaction like switches and, in some devices, head tracking is also available.

Breath-activated systems use a wide range of sensors, such as fiber optics, pressure and thermal among others, and measurements to capture the user's respiration. The signal

captured is then analysed and transformed into communication messages for supporting communication. This transformation commonly occurs in one of two ways. Discrete signal encoding consists of translating soft and heavy breathing blows to Morse code to create letters and phrases, or zero and one combinations. Continuous signal encoding involves the analysis of the speed, amplitude and phase of breathing during respiration of the respiration. This analysis creates a respiratory pattern that translates to a message [6].

Finally, there are brain-computer interfaces. These interfaces have been under research ever since the 1970s at the University of California, Los Angeles. They allow the user to control systems using their brain signals, which is very useful for people with impaired movement and speech. The interfaces used are of two types, invasive or non-invasive. Invasive interfaces involve the usage of electrodes implanted beneath the user's scalp to communicate the signals. Non-invasive interfaces use external devices, placed on top of the scalp, to monitor the user's brain. These devices use methods such as electroencephalography, magnetoencephalography and others. The signals are acquired, processed and then classified or translated. Regarding AAC technologies, the systems process the acquired signal and use it to produce communication. The process is a possible substitute for a switch or mechanical keyboard for an AAC device with indirect selection [6]. Cyberkinetics originally designed BrainGate, a brain implant system designed to help people who lost control of limbs or other bodily functions. A sensor monitors the user's brain activity and converts the signals into computer commands. Cyberkinetics, now owned by BrainGate CO., is also developing technologies for assistive communication, reporting that a user would be able to type on a virtual keyboard [16].

Discussion

Although there is no lack of technologies to support people with communication difficulties, they often present limiting factors making them unable to assist the user in all day to day situations:

- Graphical interface limitations;
- Privacy concerns;
- Invasiveness or instrusiveness;
- Sensitivity to changes in environmental conditions;
- Not suitable for the in-bed scenario

For example, most AAC technologies use graphical interfaces to interact with the user. Even though this method is common and effective, there are situations where the user either cannot get the support they need, or, in some cases, the tools were not developed to be user-centered and do not respond to the users' needs or scenarios.

In [17], Brandenburg et al. (2013) wrote a review regarding accessibility and potential uses of mobile computing technology for aphasics. Several of the articles mentioned refer the same common problems for these devices arc, i.e., . Small screens, buttons and text make interacting with the technology very difficult. This is also mentioned in [18].

The need for the user to carry a device with them, inside their home, to communicate is a limiting factor to these technologies as well. In [19], Russo et al. (2017), mentioned in their

review of high-tech AAC that the most commonly used devices consist of portable computer software, desktop computer software, and dedicated portable/desktop devices with software applications. This means that the user has to carry these devices to get the support they need. If for some reason, the user does not have the device present then no communication support exists. The solutions that employ cameras also come with issues. Although these solutions tend to be effective, some types of cameras are very much affected by environmental conditions such as lighting. Besides this, in the past few years, privacy concerns have started to be raised regarding solutions that employ cameras.

Besides all these factors, some of the solutions mentioned are also not the most effective in emergency situations. If a user has an emergency while they are not in direct contact with their device, for example during the night, it may be cumbersome or maybe even impossible to pick up the device.

The use of gesture recognition based on non-wearable, non-intrusive sensor mitigates some of the limitations mentioned. This type of solution allows the user to communicate with the help of the system at a distance and does not require the user to carry a device with them.

2.3 Gesture Recognition

Gesture recognition consists of interpreting a person's gesture through an algorithm and data provided by one or more sensors. As already mentioned above, the recognition of gestures carried out by a person can be an adequate alternative to other activation methods for supporting communication in different scenarios, such as the in-bed scenario.

The general pipeline of gesture recognition proposed by Liu et al. [20] is presented in Fig. 2.1 Most of the projects that involve gesture detection use a pipeline that is, at least, similar to the one presented.

Figure 2.1: Classic gesture recognition pipeline proposed in [20]

In the first stage, the data provided by one or more sensors are collected for further processing. There are many different types of sensors that can be used, such as wearable sensors (e.g., accelerometer, gyroscope), vison-based sensors (e.g., RGB, depth or infrared cameras), and radars (e.g., Frequency Modulated Continuous Wave radar).

The gesture identification stage refers to searching the beginning of a movement that might correspond to a gesture to recognize. Gesture tracking corresponds to tracking the movement located in the previous stage. Only the detection of dynamic gestures requires this process the latter step, as static gestures only require the processing of a single image. Both gesture identification and tracking are usually only required in solutions relying on vision-based sensor data.

The next stage is classification. During classification the gesture is matched with known possibilities. This step is normally executed with the help of a model previously trained with

a machine learning algorithm and a dataset with data from several subjects. Finally, gesture mapping takes the gesture recognition result and translates it into a pre-defined action.

Gesture recognition has evolved a lot over the years and has recently spread over multiple areas. We can see this technique present in areas such as the video game industry for interaction with the game. It is also present in the smart home area to enable interaction with the appliances and control them. It has also been explored in the automotive sector for car control to avoid the use of graphical interfaces [21].

The video game industry has used sensors such as accelerometers in the controllers, for example. One of the most famous examples is the Kinect from Microsoft. This device uses a RGB-D camera to detect, track and identify the users' actions. Wearable sensors have been frequently used to control appliances in smart homes, while cameras or radars are often used to interact with driver-assistance systems [22].

As the interest in gesture recognition has grown over the years, the list of classifiers used has also extended. Various studies investigated a wide range of classifiers, from the more classic machine learning algorithms, such as support vector machine (SVM), decision tree, random forest, and K-Nearest Neighbors (KNN), to more complex deep learning algorithms, such as Convolutional Neural Network (CNN) and Long Recurrent All-Convolution Neural Network (LRACN) [22]–[27].

As already explained above, an area where gesture recognition is present is in the video gaming industry. One of the first examples of this appeared in 2006 when Nintendo released the Nintendo Wii. The Wii was a videogame console almost solely based on gesture recognition. The user would interact with the games and the console via the Wii remote. When playing the games, the user would have to mimic throwing a ball while holding the remote when playing basketball or grab the remote as a steering wheel and then turn the controller to the side to which the user wants to steer. This technology was one of the most impactful moving forces in introducing gesture recognition to the videogame industry. As the controllers are easy and cheap to find nowadays, people decided to try and use these in different ways than what was intended.

In [25], [28], the authors took a Wii remote and built modules to use it as an interaction mean with an application. In both cases, the accelerometer sensors present in the remote provide data, which are then filtered. In [25], after filtering, a K-Nearest Neighbors (KNN) algorithm was used to cluster the data. A discrete Hidden Markov Model (HMM) was chosen due to its results in gesture recognition and a Bayes classifier is used in the final component to classify gestures. The selected gestures are "square", "circle", "roll", "Z" and "tennis". Six participants performed each gesture fifteen times, resulting in 75 gestures per participant. The average recognition rate results vary between 85 and 95%.

In [28] a low pass filter was used to remove noise from the data recieved from the Wii remote, and an idle threshold filter to remove low importance samples. A A dataset with data gathered from a single person was obtained. The training/test sets contain about 50 examples for each gesture. The gestures selected are "circle", "shake normal", "shake sideways", "down", "left", "right", "up" and "square".The model used was a HMM. The class

accuracy of recognition ranges between 80 and 100%.

In [29], the authors explore the capabilities of gesture recognition using cameras for a hand gesture recognition system to interact with a video game. The system starts by detecting the user's hand in a RGB video. The Haar cascade classifier is used to locate the hand and to classify the gestures To track the hand, CAMSHIFT and Lucas-Kanade Optical Flow techniques were used. As the tracking occurs, the system generates hand contours. The extended number of fingers is used to classify the gesture and map it to the appropriate action. Three gestures are recognized by the system: "grab", "throw", and "punch". Data from fifteen subjects was used to test the system, which obtained an accuracy of 80% for the lowest scoring gesture and 93% for the highest scoring gesture.

Another area where gesture recognition has seen a growth is the automotive area. Kopuklu et al., [21], designed a real-time recognition framework for the in-car scenario that recognizes micro-hand gestures. The authors created a dataset called DriverMHG that consists of data acquired from 25 subjects while using a simulation Logitech wheel and a Creative Blaster Senz3D camera. This camera captures RGB, infrared and depth images. For each subject, a total of five recordings were performed, each containing 42 repetitions per hand for five different gestures ("Swipe Right", "Swipe Left", "Flick Down", "Flick Up" and "Tap"), and also "Other" and "None" gestures (to help with continuous detection). The authors experimented with offline and online classification and the fusion of the different captured data types. The system achieved an accuracy of 74% online and 92% offline .

Smart-Home control is also an area where gesture detection has been employed with good results. In [30], Chou et al. developed a smart home system for appliance control and automation that relies on with multi-sensor data fusion. The system includes functions to control entertainment and household appliances, energy management, and real-time notifications for temperature, CO concentration, among others. The system contains two wearable devices. One is used to interact with the smart home, while the other helps the energy management system. The user wears the interaction device on their wrist. The device captures the data from an accelerometer and a gyroscope and uses the radio frequency module to send data to the gesture recognition algorithm. The system maps these signals to a type of gesture that controls appliances such as televisions. The signals sent by the sensors are filtered, then the mean, standard deviation, and variance are extracted and used as input for a model built with the Probabilistic Neural Network classifier. Gestures recognized by the system are: "move up", "move down", "move right", "move left", "turn right", and "turn left". The authors did not provide an accuracy metric for the system.

In [31], the authors implemented gesture recognition for smart home interaction using the gyroscope sensor embedded in a smartphone. The sensor data are converted into images, which are then used as the input for a model. The authors chose to detect six different gestures: "Horizontal grip, up and down", "Horizontal grip, down and up", "Vertical grip, up and down", "Vertical grip, up and down", "Turn right and left", and "Turn left and right". The dataset used consists of 3,805 examples, with each class containing between 483 and 828 examples. Eighty per cent of the dataset was used for training, 10 per cent for testing and

the rest for validation. The authors used transfer learning with Inception V3 for the system. The results show an 89% recognition rate.

It was decided that radar technology should be explored in the context of this work due to its decrease in cost, the good results that were obtained in different projects that used the technology, and the fact that it is non-invasive/intrusive which is a capability that is very important considering the scenario of the work. It was also selected to explore the full potential of the technology.

2.4 Radar

Radio Detection And Range (radar) is a system that uses radio waves to detect objects plus their velocity, range and angle. [32] The first reports of technology remotely similar to the radar as we know today come from 1897 when Alexander Stepanovich Popov, inventor of the antenna, noticed the reflection of the wireless signal he was transmitting. The creation of the radar has been very discussed but the credit to the invention goes to Christian Hülsmeyer [33]. According to [34], Nikola Tesla is the one responsible for bringing the invention to the US. Twenty years later, in December 1934, Robert M. Page tested an experimental 60 Hz radar that was able to track a plane 1.6 km away [34]. Over the following years, this technology would see an amazing increase in performance due to its importance in WWII as all nations involved invested in research in this area [34]. After the war was over, the technology had evolved well beyond what was thought capable [34]. This technology has evolved drastically since its early days, and radars now have higher ranges of distance and velocity. There has also been a growth in the hardware capabilities required to process the signals returned by the radar. These advancements allowed the technology to spread to various areas. It is present in aircrafts to warn of obstacles in its path, and marine vessels to measure distances to help with navigation. Meteorologists also use radar to monitor precipitation, wind and more severe weather events, such as tornados, thunderstorms and more.

This technology contains various theoretical concepts that help understand its functioning and the data it captures. For simplicity, some of these concepts are described in Appendix A.

Radar Work in Different Areas

In the past years, research and work using radar technology has led to several contributions in areas such as the automotive and health sector, the videogame industry, smart home control and gesture recognition [35]–[40]. Some of the contributions are described below.

In [35], an implementation of a gesture recognition system for the user to interact with their smart home appliances is present. The authors present a gesture recognition pipeline similar to the one shown before Fig. 2.1 based on signals captured using radar sensors. A radar sensor captures time-domain signals for the speed of the movement, for example. Initially, the authors recorded 20 repetitions for five familiar motions. These were “no movement”, “shaking head”, “nodding”, “hand lifting”, “hand pushing”. As the authors concluded that the results obtained could be improved, the Doppler values were introduced as a feature. The

results increased substantially, from a classification rate of 83 to 97% when using a 10 fold cross-validation approach.

As described in [36], the medical area has seen various uses of radar due to its capabilities of non-invasive or intrusive sensing. The radar can detect small variations in movement, meaning that its capable of sensing vital signals [37], [38]. Both articles build upon it to create systems capable of detecting heartbeat and respiratory signals by measuring the difference in phase from a signal emitted by the radar and the reflected signal. In [41], the radar was also used to track tumours during radiotherapy, leading to advantages such as a decrease in healthy tissue exposed to the radiation beam and the fact that the process requires no markers, thus being non-invasive.

Besides the medical area, radars have also been extensively used in the automotive industry. As presented in [39], the technology was used in cars to detect stationary objects such as road signs and stopped cars, or even moving targets such as pedestrians. The use of radars in this situation has advantages in comparison to cameras, as no environmental factors affect radars. Even though this approach brings benefits, it also comes with some flaws, including the problem of ghost targets and missing targets in a multi-target situation as described in [42], [40].

As mentioned before, radar has advantages when compared to other means of data capture, such as the fact that it is not affected by environmental conditions, such as light, and does not go against privacy violations, unlike most cameras. Due to this, and to its growth in the previous years, radar has become a very compelling option when it comes to gesture recognition.

2.5 Gesture Detection with Radar

In this section, the focus is on the literature more closely related with the work of this dissertation, more specifically on contributions aiming at the recognition of hand or arm gestures based on data provided by radars. A summary of these contribution is presented in Table 2.1, which includes the main relevant characteristics of each work. By examining the table, we can see that the data extracted from the radar can be used for gesture recognition with good results.

In [44], presents a review of some of the types of radars used and their applications. From the different available radar types, the focus in this section will be on Frequency-Modulated Continuous Wave (FMCW) radars, since this type of radar has been vastly used in for gesture recognition, making it a viable option for this work. An FMCW radar emits a continuous wave with a varying frequency over time, allowing to obtain information on the position and velocity of moving targets. More information about this kind of radar can be found" in Appendix A.

One of the contributions mentioned in [44] reports to have used a FMCW radar to create a driver assistance system [22]. The authors of this contribution used a multi-sensor approach, which consisted of a FMCW radar, a RGB camera, and a depth camera. This use of mixed sensors makes the presented system very robust when dealing with environmental conditions

Ref	Radar	# subjects	# gestures	# examples per gesture	Classifier	Accuracy (%)
[22]	FMCW	3	10 plus random hand motions	10-20	CNN	94
[23]	FMCW	5	4	50	Random Forest	92
[26]	IR-UWB	—	6	—	NN[1]	99
[43]	FMCW	3	3	1000	CNN	99
[24]	FMCW	10 training, 5 testing	5	150 training, 600 testing	LRACN	98

[1] Type of neural network was not specified.

Table 2.1: Summary of the contributions analysed in this section, including information on the used dataset and classifier, as well as the obtained accuracy for gesture recognition.

such as light. The radar in this system retrieves range and Doppler data. The authors assume that a gesture occurs when the radar detects motion. After the radar detects a gesture, the cameras switch on. The behaviour described occurs because the radar requires less power when compared to the cameras, making the system more power-efficient.

After capturing the gesture, the data obtained are filtered and the depth and RGB images are used to obtain the hand region mask, which is then normalized to the range [0, 1]. The authors use the depth camera images to obtain the hand region mask. The values of the region go through a normalization process to a range of [0, 1]. The image from the colour camera goes through a similar process, converting an RGB image to the same range of values as before. Next, the radar data are mapped to the depth data, allowing the authors to extrapolate the angular velocities for the depth images using Voronoi tessellation. After overlapping the hand region mask with the previous image, the authors obtained the angular velocity values for the hand region. All resulting images are resized for model training. The classifier used to build a model for recognition of dynamic gestures is a convolutional Convolutional Neural Network (CNN). This network consists of two 3D convolutional layers, which automatically learn discriminatory spatiotemporal filters to reduce the dimensionality of the input gesture data. The authors recorded a dataset that consisted of a total of 1,714 gestures executed by 3 subjects. The gestures included "left/right/up/down" palm motion, "shaking" of the hand, "Clockwise/Counter-clockwise" hand rotations, "left/right swipe", and "calling". These were recorded both outdoors in a real car and indoors in a driving simulator. The best overall performance of the system achieved was an accuracy of 94.1% with a combination of all three sensors.

In [23], the authors present the system *Soli*, a high-resolution, low-power and compact gesture sensing technology for human-computer interaction, based on a FMCW radar. In this work, the authors focus on exploiting the temporal accuracy of the radar, as gestures can be recognized directly from temporal variations in the signal. The authors also focus on: maintaining a high throughput of the sensor to minimize latency; exploiting the advantage of using multiple antennas to reduce noise, maximizing Signal to Noise Ratio (SNR); providing continuous and discrete gesture recognition; and being computationally efficient. The authors

claim that some features are extracted due to intuition and radar knowledge, and that the most relevant features are automatically detected and selected during the learning phase. The authors choose to use the random forest algorithm for recognition, due to it being effective and fast. A Bayesian filter was used to process the raw predictions, decreasing false positives. In total, the dataset included 2,000 gesture examples (5 participants $\times$ 4 gestures $\times$ 50 repetitions $\times$ 2 sessions) with a 1000 examples used for training and testing. The gestures are the "virtual button", "virtual slide", "horizontal swipe" and "vertical swipe". Accuracy values of 87% and 92% were achieved when using unfiltered and filtered predictions, respectively.

In [26], present a proof of concept of a gesture recognition system using an impulse radio ultrawideband radar without being applied to a given context. The raw signal captured by the radar is filtered to remove background noise. Both the unfiltered and filtered signals are used in the classification process to improve accuracy. The system uses two machine learning techniques, the unsupervised Principal Component Analysis (PCA) technique to extract features from the signal, and supervised learning using a neural network to classify gestures. PCA allows the authors to use only the ones with the most variability and therefore avoid overfitting the model and reducing the model complexity and size. The authors selected six gestures for detection , one being a control gesture where no movement occurs. The training and test sets consisted of 2,200 and 500 examples, respectively. The authors report an accuracy of 100%.

In [43], the authors take a more usual approach to the work. Three hand gestures are distinguished, and for each of them 1,000 spectrograms were generated from micro-doppler values. The distinguished gestures are "beckoning", "swipe" and "wave". The authors decided to use a CNN for classification, as these have proven to be very effective. The CNN used consists of 10 layers and two input channels corresponding to the real and imaginary part of the spectrogram to provide both amplitude and phase information for gesture recognition. The training, test, and validation sets contained 90%, 5% and 5% of the dataset (900, 50, and 50 spectrograms per gesture), respectively. After training and validation, the authors obtained a classification accuracy of 99% over the test set.

In [24], the authors propose a short-range gesture recognition system using a millimetre wave radar. Range-Doppler images are created and used as input for the classification model. Each pixel's intensity in these images represents the reflected energy from each point. For feature extraction and classification, the system uses a Long-Term Recurrent All-Convolutional Network. In this type of network, fully connected layers are replaced with convolutional layers to mitigate the effects of overfitting when using a small dataset. The authors decided to use five gestures for detection "grab", "finger rub", "swipe", "up-down", "circle". Ten subjects recorded 150 sequences. Due to the limited size of the dataset, the authors used data augmentation techniques. These techniques create synthetic images with variance to the originals to resolve generalization problems. The network used a training-validation split of 80%-20%. Regarding the test set, it included 600 sequences acquired from five different subjects to use as a testing dataset. The obtained results show an accuracy ranging from 85 to 98%.

2.6 Conclusion

This chapter presented some background and a revision of the state of the art in the areas relevant to this dissertation, including communication disorders, aphasia, radar and gesture recognition. The analysis of this information was essential to understand the technologies and the target users and served as the basis for the work presented in the next chapters.

For example, it was possible to conclude that although many assistive technologies for communication disorders already exist, these still face some limitations and challenges, such as privacy concerns, invasiveness or lack of versatility. Either the user is required to carry a device with them, or the solution is not versatile enough to be used at all times, or it is cumbersome to use.

As an alternative to the existing solutions, the use of gestures allows a more versatile and natural way for supporting communication at a distance. As for the sensor to be used for recognizing the gestures, an unobtrusive sensor, such as a radar, revealed to be the most appropriate solution, due to being low-cost, unlike RGB cameras it does not require light to function and is not affected by other environmental conditions.

CHAPTER 3

Personas, Scenarios and Requirements

Even though the work is exploratory, it was important to get to know the problem better and approach the research, as best as possible, from a human-centred perspective. This required the help of some literature concerning communication disorders, as well as the consideration of methods supporting a human-centred approach. As such, the use of tools like personas and scenarios allows the identification of the needs of people with those disorders, leading to the creation of better requirements. Additionally, it serves as a brief illustration of the envisioned scenarios for the work. They also help make sure that the work is moving in the right direction when it comes to assisting people with communication difficulties.

3.1 Personas

Personas describe the target user for the work. This tool helps us understand the needs and behaviours of the users.

In the following paragraphs, we describe three different personas, scenarios and motivations. Each one of these depicts a different aphasic person and different scenarios that they face.

3.1.1 Persona 1: João

João: Male, 44 years old. João is a 44-year-old male from Aveiro. João has worked as a teacher in Universidade de Aveiro for 15 years. Outside of work, João has a very active social life and often goes out with friends. Due to his profession and social life, communication is a crucial part of João's life and is something that he enjoys. Around a year ago, João was the victim of a stroke that left him with aphasia. The condition left João with serious difficulties communicating. As communication was such a vital aspect of João's life, these difficulties took a toll on his self-esteem, leading him to a more isolated lifestyle.

Motivation: João wants to regain some of his communication abilities, and to do so requires some help communicating.

3.1.2 Persona 2: Raquel

Raquel: Female, 50 years old. Raquel is a 50-year-old female from Aveiro. She lives alone as her son, Luís, has grown up and moved out of the house. Raquel worked for 25 years as a chef in a local restaurant. She has always lived a very active life and has taken care of herself ever since her son moved out. This busy life includes doing groceries, cleaning the house, among other things. A few months ago, while doing groceries, Raquel tripped and hit her head. Unfortunately, this accident affected her brain resulting in aphasia. The disorder takes a toll on Raquel's day-to-day life as she no longer has the confidence to live the active life she led before.

Motivation: Raquel wants to regain her independent lifestyle and, to do this, she needs to get some confidence back. As she has trouble communicating, this is a difficult task.

3.1.3 Persona 3: Rui

Rui: Male, 60 years old. Rui is a 60-year-old male from Aveiro and has worked for 30 years as a mechanic. Rui lives with his wife Rita, who is 64 years old. They live alone in a small house in the city. They have been married for 27 years and are still very much in love. A year ago, Rui was in a car accident. Due to complications regarding a concussion, Rui developed aphasia. The communication difficulties that arise from the condition have taken an impact on Rui's day to day life and his relationship. Due to the illness, Rui and Rita have a lot of difficulties communicating with each other. These difficulties have taken an impact and left both with an extreme sense of anger and sadness.

Motivation: Rui wants help communicating so that he and his wife can recover some of their old lives back, and improve his quality of life.

3.2 Context Scenarios

Context scenarios describe situations where, in this case, a user requires support or assistance. These scenarios describe contexts where a user interacts with the system we envision. They help understand the main focus that the system should have and how its features should assist the user.

Different possible scenarios are described below. For each scenario, it is indicated which personas are involved, and information of where, when and how the scenario takes place is also provided.

3.2.1 Scenario 1: João

Where: Bedroom.

When: During a friend's visit, in the afternoon.

How: With his friend, both in the bedroom.

To get some assistance while communicating, João decides to install the system in his bedroom. When his friends or family come to visit João, he is finally able to communicate better. During a visit, João's friend asks him a question. To help express his answer, João

performs a gesture which the system recognizes and translates into a message, which is presented using speech output (relying on a speaker included in the system) to audio output.

This kind of assistance allows João to have an easier time maintaining and participating in a conversation, allowing him to regain some of his old social life.

3.2.2 Scenario 2: Raquel

Where: Inside the house.

When: Afternoon. Evening

How: Lying in bed, carer in the living room.

After Raquel's accident, her inabilities to communicate forced her to live with a carer. As Raquel led an independent life before the accident, living with a carer is a challenge. One day Raquel decides to install the system in her apartment. The interaction with the system allows for an easier time expressing her needs and wants to her carer. One afternoon, while in bed, Raquel uses the system to signal that she requires assistance.

In this situation, the system helps Raquel during the interaction, thus making the whole process easier.

3.2.3 Scenario 3: Raquel/Luís

Where: Inside the house.

When: Night.

How: Raquel lying in bed, alone. Luís outside the house.

One night, while lying in bed alone, Raquel starts feeling discomfort in her chest. After a while, the irritation turns to pain, and Raquel realizes that she needs assistance. With this realization, Raquel signals the system that she requires assistance. After interpreting the gesture, the system sends a message to Raquel's son Luís, who calls an ambulance and then goes to his mother's house.

In this situation, the system allows for a much easier interaction in case of an emergency. Using other kinds of devices is complicated or sometimes impossible in emergencies. Raquel and Luís realize that the system can help her in emergencies, leaving both more at ease and allowing Raquel to be more independent.

3.2.4 Scenario 4: Rui/Rita

Where: Inside the house.

When: Morning.

How: Rui Lying in bed, with his wife Rita.

Sometime after the accident, Rui decides to install the system in his bedroom. One morning, Rita asks Rui what he would like to eat for breakfast. Even though this is a simple question, Rui has a troublesome time responding due to his difficulties. However, with the system's help, Rui can now express his wants and needs to his wife and is able to answer her questions.

This help is very valuable to Rui, as having a conversation with his wife is something he loves. Without the system, the conversations are short, complicated and even frustrating for both parties.

3.3 Requirements

The personas and context scenarios depicted previously allowed for the creation of the necessary requirements for the work. These requirements fall into two categories, functional and non-functional. Functional requirements define what the system requires to work as intended. Non-functional requirements focus on the main features of the system, regardless of how they should be implemented. In this case, non-functional requirements dive into the interaction between user and system.

3.3.1 Non-Functional Requirements

- **Gestures that are simple, easy to perform and remember:** The system supports people with communication problems. One of the leading causes of aphasia are strokes. When selecting gestures for the system to recognize, it is necessary to consider this fact. The movement needs to be simple enough to execute to ensure that even people who suffer from aftereffects from a stroke can do it. The selection of gestures also needs to consider that the user is in bed while performing the movement. Finally, as the user will use the system every day and maybe in emergencies, the gestures should be easy to remember to minimize response times.
- **Appropriate feedback after gesture execution:** After executing a movement, the system needs to provide feedback to ensure the user's command is recognized. This feedback can be a sound or a speech generated sentence.
- **Non-invasive and non-intrusive:** The system should not be invasive for the user, i.e., it should not require the user to carry or wear a device to interact with the system, and it additionally should not put their privacy at risk.
- **Suitable for the bedroom scenario** - The system should be easy to deploy and used in the context of the bedroom, more specifically when the user is lying in bed, whether it is during day or night-time.

3.3.2 Functional Requirements

- **Appropriate technology for the system:** Based on the non-functional requirements mentioned above, the system should be non-invasive and non-intrusive and capable of being used at night or during the day. Given these requirements, the use of radar technology is the best option.
- **Suitable radar configuration for the scenario:** The radar detects various degrees of motion at different distances, making it a versatile tool. However, to achieve better results, the configuration in use needs to be appropriate for the considered scenario i.e., the user's bedroom.

- **Processing unit capable of all processing stages:** It is required to have a processing unit capable of acquiring the data provided by the radar and transform them into features that are useful for gesture classification.
- **Model capable of radar-based gesture recognition:** The system requires a model that is capable of recognizing gestures from the data sent by the radar. To obtain that model, machine learning techniques can be used (including deep learning and transfer learning). Model training also involves a dataset gathered from one or more subjects while they carry out the gestures to be recognized.
- **Continuous operation during both day and night:** The system should be able to work continuously during both day and nigh to provide round the clock support to the user. For this, the choice of radar and power supply are paramount.

3.4 Conclusion

Even though the work is exploratory, we felt the need to explore the literature around communication disorders and aphasia. This literature helped gain a better understanding of the problems faced by the ones afflicted by these problems. The gained knowledge allowed for the creation of personas and scenarios that demonstrate some of those challenges. These scenarios show the situations where the solution described could help people. The personas and scenarios helped shape the main requirements for the work. They helped define the main characteristics of the system. The non-functional requirements also helped defined the functional requirements, such as the choice of technology to use.

CHAPTER 4

Radar-Based Gesture Recognition System

This chapter presents the proposal of a system for gesture recognition that aim at aiding communication when the user is alone in a bedroom, lying in bed, and may need to communicate with other people (e.g., caregiver, family member) to ask for help, for example. It depicts the design and development of said proof of concept.

4.1 System Overview

As a proof-of-concept, we implemented a first prototype that relies on the setup shown in on the left side of Fig. 4.1, which includes a bed and a radar, in this case, a AWR1642 FMCW radar from Texas Instruments. The radar is elevated 0.55 m from the ground, and placed at 1 m from the bed, on the left side of the subject, parallel to the longest side of the bed. The radar's 2D coordinate system is shown in Fig. 4.1.

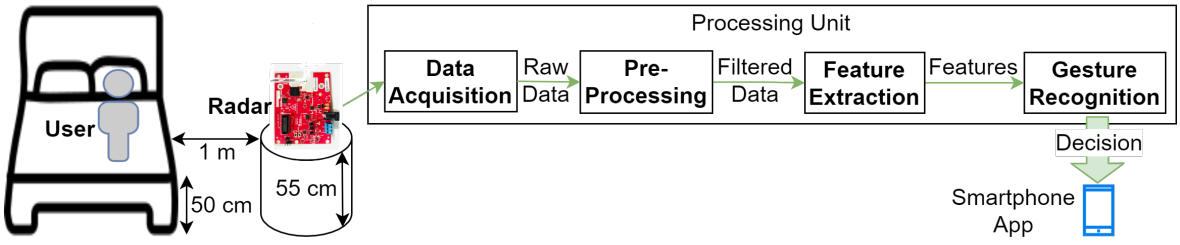


Figure 4.1: Architecture of the proposed system, including a possible setup for the bed and radar, as well as the pipeline for gesture recognition.

The overall architecture of the system is depicted in Fig. 4.1. A radar captures data from the detected moving targets, in this case the human body. These data are sent to a processing unit, where they are pre-processed by removing outliers. Features are then extracted and used to recognize the gesture being carried out. After classifying the gesture, the system maps the gesture to a notification, which is sent to a smartphone.

For the processing unit, any computing device with data processing capabilities is usable. In this prototype, we used a computer, but this can be replaced by a device such as a Raspberry Pi for the deployment in real scenarios. The radar connects to the processing unit via USB. For the radar we used version 2.1 of the MMWave SDK and the Demo Visualizer application to capture the data. To extract the data, we used a Matlab script. The Matlab version used was R2020b.

4.2 Gestures

The gestures that the system is able to recognize are listed and described in Table 4.1, where only the four first gestures are meant to be used for supporting communication. These involve the use of one of the arms and were selected aiming at simplicity and based on initial feedback from therapists and carers on the gestures' suitability for aphasic patients lying in bed. Moreover, they can be used for generating simple messages (e.g., I need help") and "Yes/No" answer. "Rotate" and "Knock" were also chosen due to the interaction between the movement and the environment. "Rotate" is also considered a "no gesture" as we are only detecting arm movement. This gesture was recorded due to being a common movement to happen as the user is lying in bed.

Table 4.1: Arm gestures considered for the system's prototype. All gestures begin with the subject's arms resting on the bed, extended and parallel to the body.

Gesture	Description
Wave	Move the arm and hand from left to right and back, starting with the arm parallel to the body.
Raise Arm	Raise the arm until a 90º angle is formed with the body and then lower it back to initial position.
Back and Forth	Move the forearm towards the arm making an angle below 90º, starting with the arm extended, parallel to the body, and returning to full extension.
Knock	Move the forearm towards the arm making an angle around 90º, and then lower it down, knocking the mattress.
Rotate	Starting with both arms resting on the bed, rotate your body towards one direction and back.

4.3 Radar Configuration

Before using the system, it is necessary to define a suitable radar configuration for the scenario. Several configurations were tested with the goal of obtaining the best configuration for the in-bed scenario. The testing of the configurations was done using a simplified scenario of the bedroom scenario. The configuration chosen was the one that had the best ratio of information and noise. Table 4.2 contains the chosen values for each parameter used in the proof of concept.

Table 4.2: Radar configuration parameters

Parameter	Value
Scene selection:	
Frame rate (fps)	20
Range resolution (cm)	4
Maximum range (m)	10.28
Radial velocity resolution (m/s)	0.22
Maximum radial velocity (m/s)	3.47
Object detection:	
Range direction	NO
Doppler direction	**YES**
Remove static clutter	NO

As we can see in the first section of the table named "Scene selection", the radar was configured to capture data with a sampling rate of 20 fps (frames per second). Targets with a maximum radial velocity of 3.47 m/s are detected up to a distance of 10.28 m. In this configuration, range and radial velocity resolutions are 4 cm and 0.22 m/s, respectively, where these resolutions refer to the minimum distance and radial velocity required between two targets for the radar to be able to distinguish them.

The second section of Table 4.2, named "Object detection", refers to the peak grouping capabilities of the radar. Peak grouping is the ability to report a single point instead of a cluster of neighboring points. Range direction and Doppler direction differ on the criteria for the clusters. The first focuses on points with the same range direction, while the second focuses on Doppler direction. If one of these options is selected, then the radar detects a cluster of points and only considers the one with the highest relative power detected. As we were trying to reduce the amount of pre-processing done to the data by the radar, the option "Remove static clutter" was disabled. This was due to the fact that we did not want the radars processing to remove any information that might have been useful.

4.4 Data Acquisition and Preprocessing

When recording, the radar captures movement and sends this to a processing unit in the form of packets. The data from the radar packets is extracted, filtered and is transformed into grayscale images. These images are used as the inputs for the system.

Each data sample provided by the radar includes three different data types:

- X Coordinate
- Y Coordinate
- Doppler Index

The X and Y coordinates correspond to the distance between the target and the radar, in the coordinate system shown in Figure 4.1. The Doppler index gives us information about the movement, with the value of the index depicting the intensity of the movement and its sign depicts the direction of the movement.

The data from the radar comes in the form of packets, which contain a header and the Tag Length and Values (TLV)s, depicted in 4.2 and 4.3.

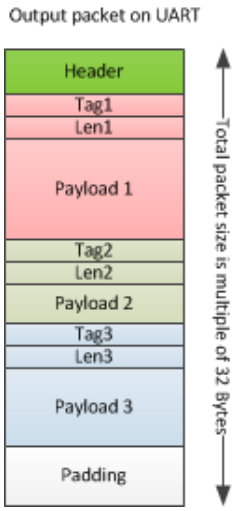


Figure 4.2: Description of the output packets. Image taken from Texas Instruments technical documents

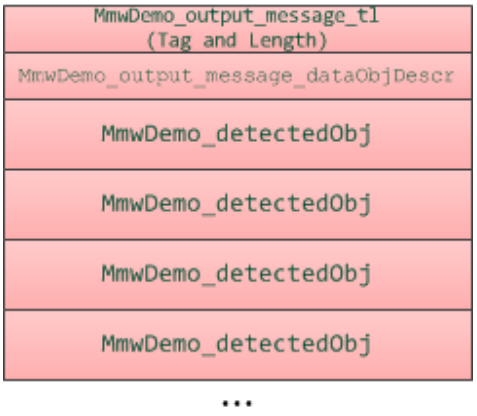


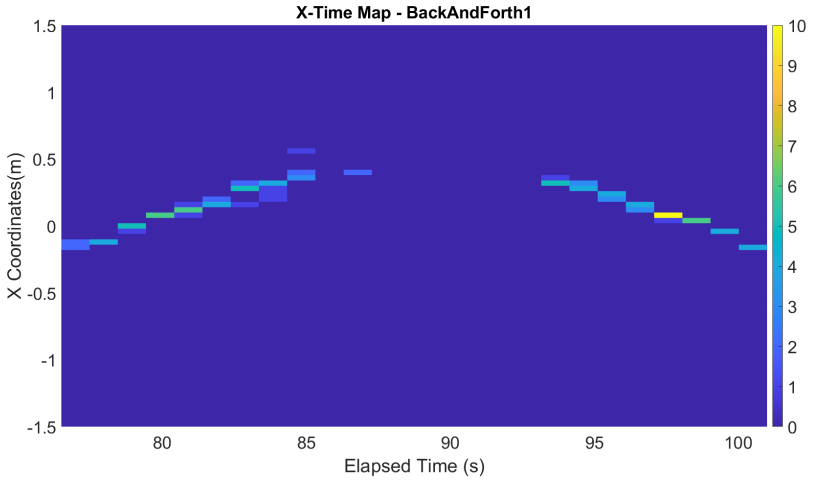
Figure 4.3: Description of the detected objects field. Image taken from Texas Instruments technical documents

The acquired data are processed using a sliding window of 5 s without overlap. For each window, pre-processing consists of removing outliers corresponding to unwanted reflections or noise. A detected target is considered as an outlier if its Euclidean distance to the radar is outside a selected interval or its velocity is very close to 0 m/s. All data with X and Y coordinates outside given intervals were also discarded. These intervals where chosen according to the setup.

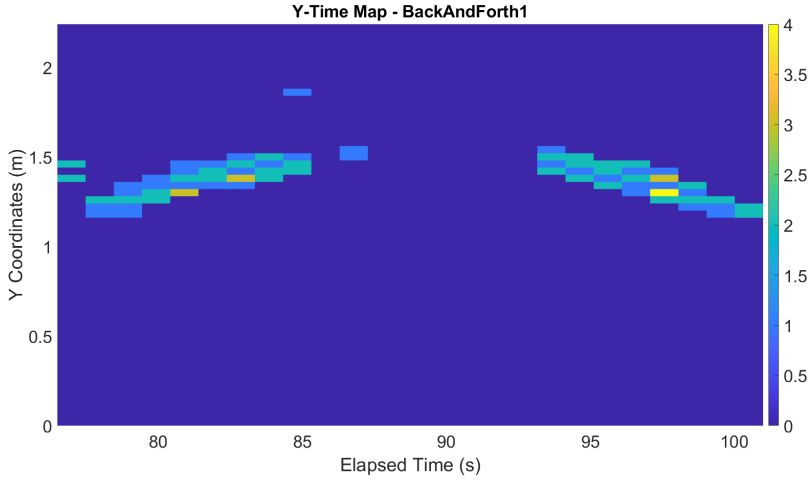
4.5 Feature Extraction

The feature extraction step corresponds to generating three different maps from the filtered data, one for each data type versus the elapsed time (X-Time, Y-Time, and Doppler-Time maps). An example of the X-Time, Y-Time and Doppler-Time maps for a repetition of "Back and Forth" is presented in Figs. 4.4a,4.4b, 4.4c respectively, where the colour represents the number of detected targets (bright yellow corresponds to the maximum value for each map, while dark blue corresponds to no detected target). Please note that the beginning and ending of the feature extraction window where no movement is detected are discarded.

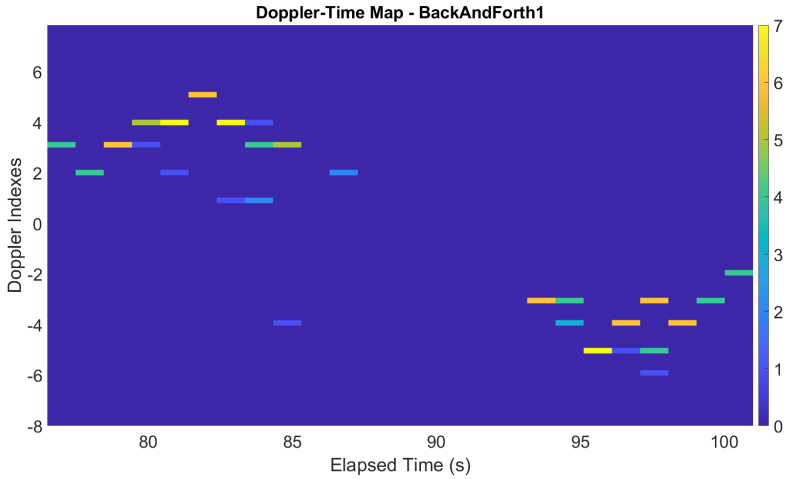
The matrix associated to each map is used to obtain a normalized greyscale image. The three images are then combined into a single image (X-Time above Y-Time above Doppler-Time). This combination is done in order to use all three data types as input for the model. Figure 4.4 shows examples of three combined images for gestures "BackAndForth" (a), "Knock" (b) and "Wave" (c), respectively.



(a)



(b)



(c)

Figure 4.4: Example of the X-time (a), Y-Time (b) and Doppler-time (c) maps for a repetition of the "Back and Forth" gesture performed by a given subject.

(a) (b) (c)

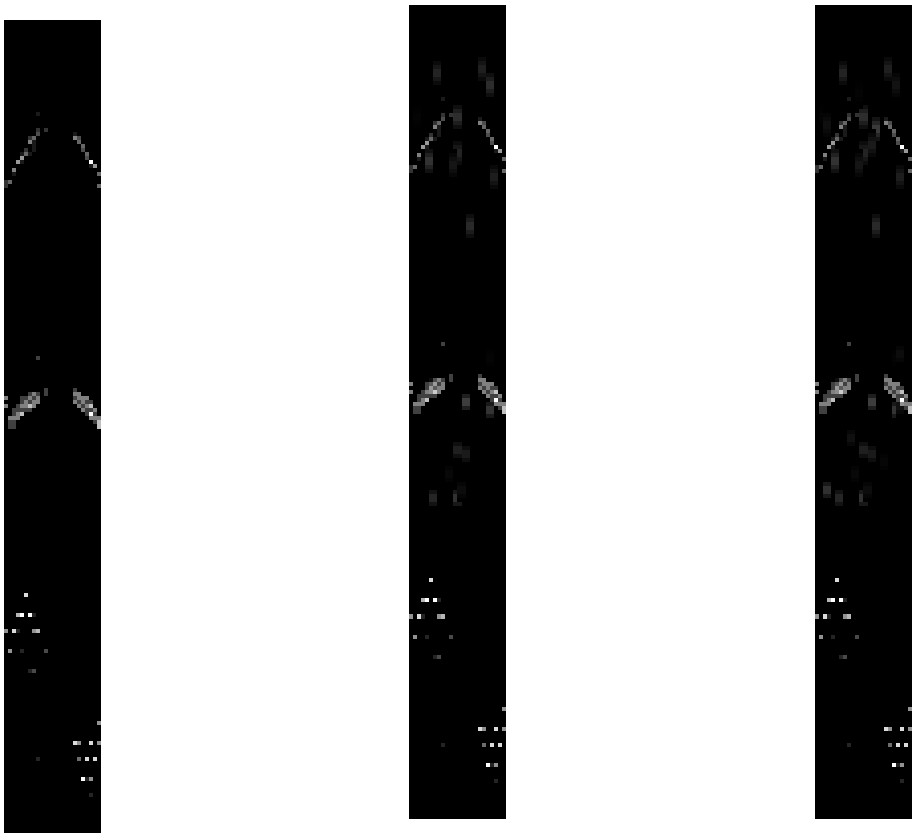
Figure 4.5: Examples of combined images for gestures "BackAndForth" (a), "Knock" (b) and "Wave" (c).

4.6 Gesture Recognition

The images resulting from feature extraction are fed into a model that performs gesture recognition. Several classifiers have been tested and used. This model is previously trained using the transfer learning method, relying on a pre-trained deep neural network model for image classification, and a given dataset with gesture data collected from one or more subjects. After analysing the results in 5.12, the most appopriate model for the scenario is MobileNetV2, which was used for all the tests performed in section 5.4.

Due to the restrictions imposed by the COVID pandemic, it was not possible to gather a large dataset. To mitigate this problem, it was decided to use a technique called offline data augmentation. Data augmentation consists of increasing an existing dataset by generating a larger number of lightly modified copies of the original dataset. In this case, the technique

modifies the initial images by adding noise to obtain new ones. Two different types of noises were considered for augmentation. The first uses classic noises, such as Gaussian, Poisson, and salt and pepper. For each image, a random noise or a combination of the three is chosen randomly. The second approach creates "noise blocks" with random intensities and locations. This type of augmentation was created for this work to try to simulate the presence of a target other than the user's arm or a reflection detected by the radar.



(a) Example of an image from the original dataset

(b) Example of the original image augmented with classic noises

(c) Example of the original image augmented with generated noise blocks

Figure 4.6: Comparison between original and augmented images.

The language selected for the classification process is Python due to its capabilities in data processing for machine learning and various available packages that facilitate the process. The pre-trained models used for transfer learning are available with libraries Keras, and the data processing and organization functions used were based on Tensorflow and Python.

4.7 Conclusion

This chapter presents a prototype of a system for gesture recognition based on radar technology with the aim of assisting communication for people with difficulties. The prototype recognizes four gestures based on arm movement, and a "no gesture", which is a common movement for a person lying in bed. The criterium for choosing the gestures was the ease of use while lying down in bed. The radar captures information such as the X, Y coordinates

and Doppler indexes of the detected targets. A filter sifts the data and removes all the unwanted reflections and noise. This filtering process consists in removing all points that have a Doppler index close to zero. As the Doppler index represents the intensity and direction of the movement, if a target has a Doppler index of zero, it is not moving and thus not a part of the gesture. The sifted data should now only contain information regarding the subject's movement. Next the data is transformed into images. After creating images for each data type, these are grouped into a single one. These combined images are the input for the model.

CHAPTER 5

Results

The development of the system presented in the previous chapter went through various stages during its lifecycle. All the different stages and experiments are depicted throughout this chapter.

5.1 Radar Exploration

The first stage consisted in understanding how the radar worked and getting acquainted with the format of its data packets. This theoretical knowledge required a lot of research and reading, meaning that it took some time before the radar was up and running.

After understanding the data and how the captures work, it was necessary to extract the different samples and data types from the packets. A Matlab script was created to extract the data from each capture. For the data extraction from packets to work correctly, it was necessary to know the exact size of each component of the packets. Meaning that, once again, some research was required.

After analysing the literature needed, the first experiments were carried out. These involved capturing data and analysing its components and creating different visualization forms to search for patterns to see if the approach was viable for gesture recognition. The following section goes more into detail about these experiments.

5.2 Preliminary Experiments

5.2.1 Data Exploration

After the software to capture radar data was up and running, the first data capture was made and it was finally possible to extract the data and start the preliminary experiments. Three gestures were recorded, "BackAndForth", Wave and Raise Arm. Now it was necessary to start testing the data to see if the project's goals were attainable. The first experiments consisted of data exploration to find any visible patterns. The following figures 5.1, 5.2 show the gathered data in the form of a scatter plot.

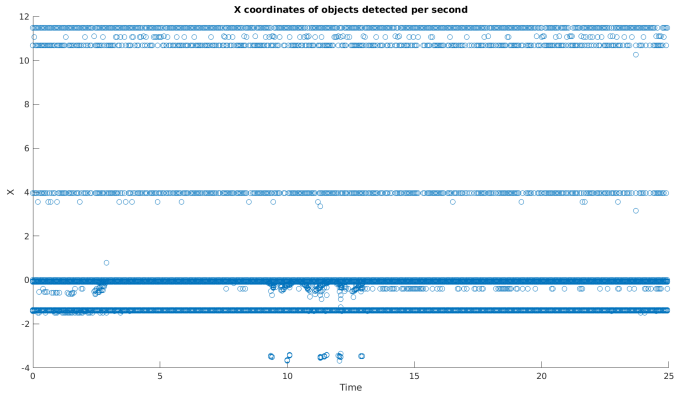


Figure 5.1: Unfiltered X coordinate (in meters) versus the elapsed time for the "Back and Forth" gesture.

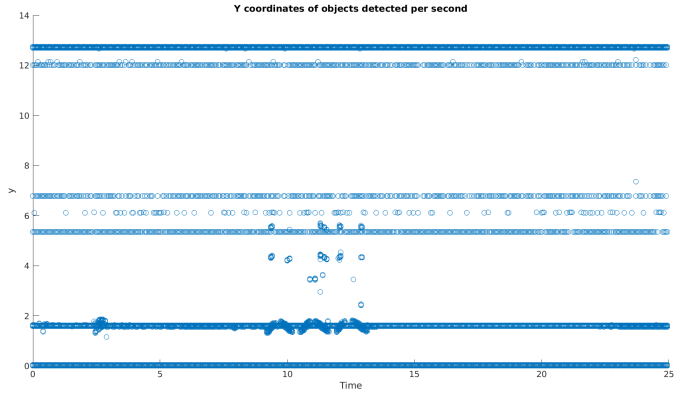


Figure 5.2: Unfiltered Y coordinate (in meters) versus the elapsed time for the "Back and Forth" gesture.

Looking at the images we can see that the data contains a large number of detected targets. As some do not vary their position it is safe to assume that these targets are unwanted noise and reflections that are unnecessary. As the data contains a lot of unwanted noise and reflections that may make it harder to distinguish between different movement patterns. Therefore, The data needs to be filtered. This filtering consists in removing points that are not moving. These points will have a Doppler index outside of interval [1e−5, 10] and will be removed. In all experiments described in this chapter, the filtering process also included the removal of points outside of the expected of [-1.5, 1.5] m and [0, 2.25] m for X and Y coordinates, respectively. The only exception is the last experiment, where filtering relies only on on the Doppler index.

The following figures 5.3 and 5.4 shows the same data after being filtered. As most of the noise was removed, it is much easier to see the arm movement and the corresponding patterns.

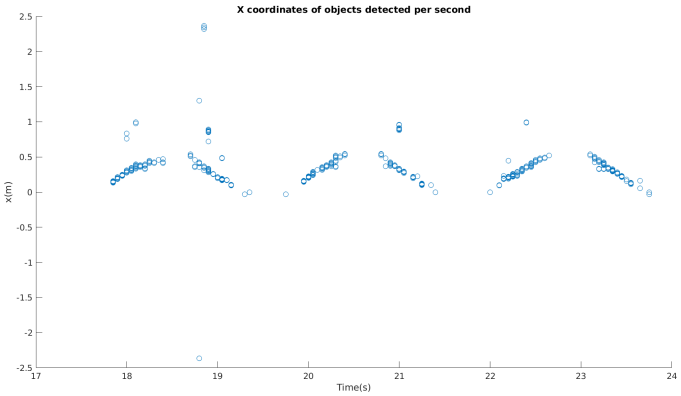


Figure 5.3: Filtered X coordinate versus the elapsed time for the "Back and Forth" gesture.

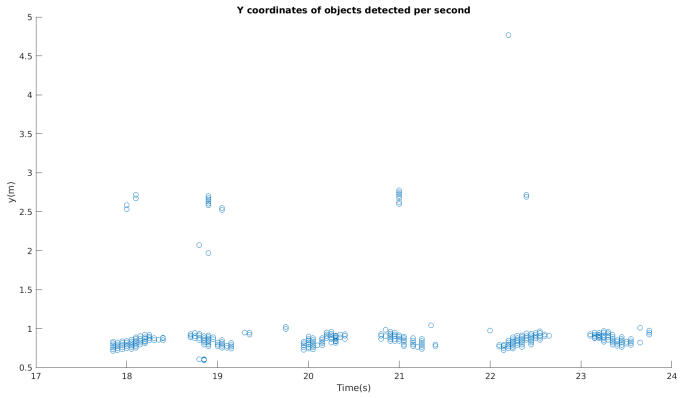


Figure 5.4: Filtered Y coordinate versus the elapsed time for the "Back and Forth" gesture.

After establishing that the data extracted from the radar contains different patterns for each gesture, it was time to start experimenting with classification. In this next preliminary experiment, the first experiment used a classic approach to classification.

5.2.2 First Experiment with Classification

After establishing that the gestures generate different patterns between each other, it was time to experimenting with classification. For this experiment a pipeline from data capture to testing was created. The complete implemented pipeline with all the stages is present below.

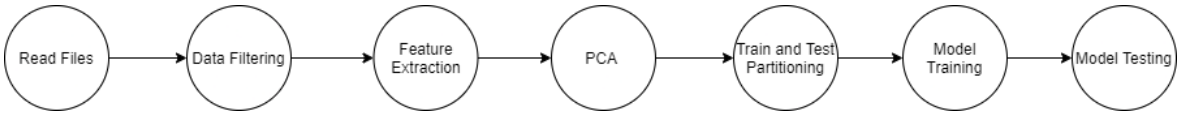


Figure 5.5: Scheme of the classification pipeline.

The first two stages have already been described in chapter 4. The following stages are described below.

Seven features are extracted from the signals corresponding to a gesture repetition, for each data type (i.e., X coordinate, Y coordinate, Doppler index): minimum, maximum, mean,

median, standard deviation, variance and covariance.

As seven features for each of the three data types generates a large number of data to use in the model, we decided to use PCA. Principal Component Analysis is a technique that allows us to reduce the dimensionality of the dataset while maintaining most of the information. The technique transforms large sets of variables into smaller ones [45]. PCA is helpful as dimension reduction allows for a smaller load on the model while preserving variation in the data.

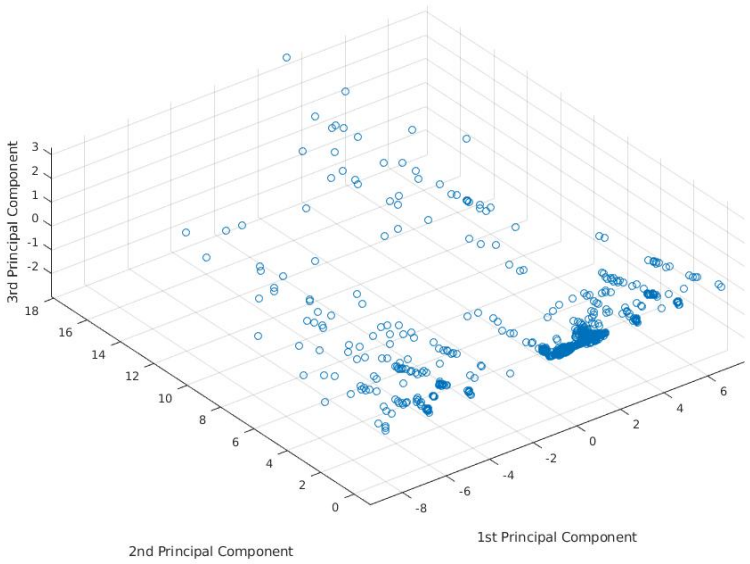


Figure 5.6: 3D Plot showing the variation between the 3 principal components.

```
explained =

   64.674
   32.598
    1.492
   1.1063
0.080116
0.050036
2.94e-31
```

Figure 5.7: Explained variance of the features where PCA was applied. As we can see, the first three contain over 98% of the variance in the data

As we can see in Fig. 5.6, the first two components show a lot more variability than the third. It is also possible to see that even though the first two have approximately the same variability, there are more points scattered along the first component axis, meaning that this component will account for more variability in the data. These conclusions are verified by Fig.

5.7, which shows us the variability of the components present in the data. As we can see, the first three components account for more than 98% of the total variability.

After reducing the dimensionality of the dataset, the resulting dataset was used to explore the possibility of using a classic classifier for distinguishing between different types of gestures. K-Nearest Neighbors was the classifier used in this preliminary experiment,due to its simplicity. To obtain the best possible model, we tuned the classifier's hyperparameters during its evaluation. The hyperparameters tuned are described in Table 5.1. Hyperparameter tuning consists in choosing the best hyperparameter for a classifier. This process was automated and we used function "RandomSearch" for Matlab.

Several parameter combinations were tested, but none led to an accuracy value over 65% as present in Table 5.2.

Parameter	Definition
Distance	Defines the distance metric used by the algorithm.
Weighted Distance	Defines whether to use a distance weighing function and which one.
Exponent	This parameter is only used if the algorithm uses the "Minkowski" distance.
Number of neighbours	Defines the number of neighbours to find for classifying each point when predicting.
Standardize	If true, then the software centres and scales each column of the predictor data by the column mean and standard deviation, respectively.

Table 5.1: Description of the hyperparameters for KNN algorithm.

Numb. Neighbors	Distance	Distance Weight	Exponent	Standardize	Accuracy (%)
116	Minkowski	Inverse	0.67337	True	63.4
107	Chebychev	Squared inverse	N/A	False	59.7
150	Euclidean	Squared inverse	N/A	True	61.1
128	Euclidean	Inverse	N/A	True	62.8
136	Chebychev	Inverse	N/A	False	64.9

Table 5.2: Accuracy achieved for KNN and 5 test runs when using the for the random search optimizer for hyperparamter tuning.

After having the best possible combination of hyperparameters it was time to partition the dataset into train and test sets. This was a 80/20 split with 80% of the data being used for training and the rest for testing. The results obtained were not promising, so it was decided to try a new approach using Dynamic Time Warping, to further explore the data in a more visual way.

5.2.3 Experiments with Dynamic Time Warping

It was decided that it was best to start this stage with a simple technique, so we chose Dynamic Time Warping (DTW) due to its simplicity and that it allows us to calculate the similarities between timeseries which vary in speed. The Python library used is called fastdtw. After obtaining the vectors with the doppler indexes, each of the vectors was compared with another. This process was repeated for all vectors. It was decided to use doppler indexes as this is the data type where the differences in the patterns were more visible. Figure 5.8 represents the comparison between the Doppler vector of two different "RaiseArm" captures , in terms of Euclidean distance obtained with DTW.

Figure 5.9, shows the result of the comparison between captures of "RaiseArm" and "Wave".

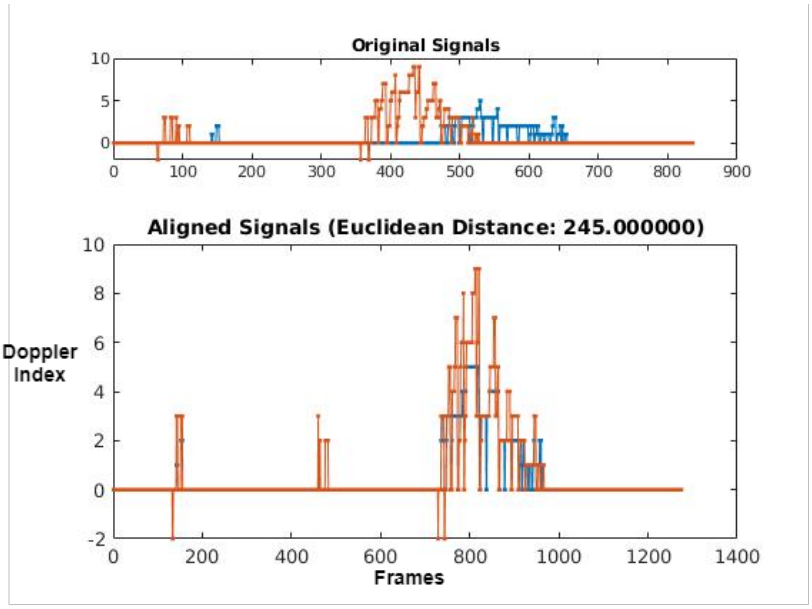


Figure 5.8: Distance between Doppler indexes from different "RaiseArm" captures.

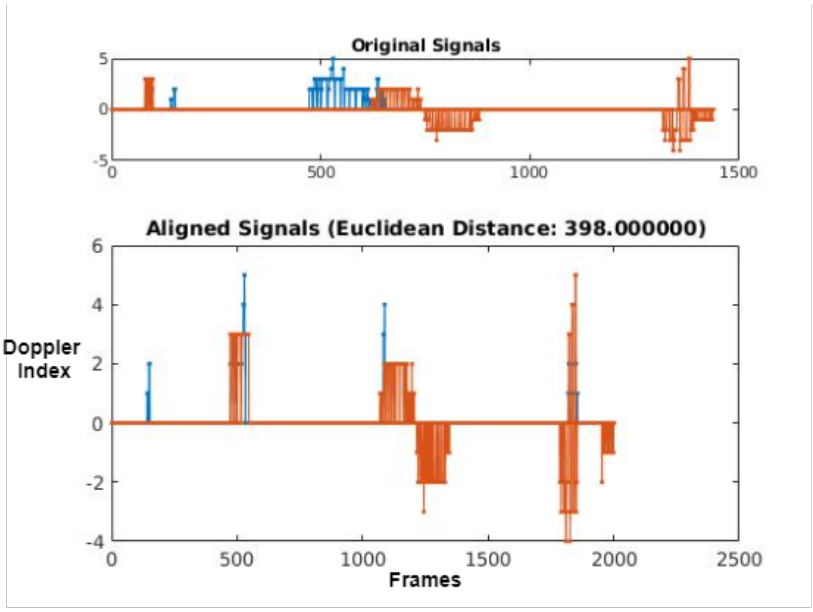


Figure 5.9: Distance between Doppler indexes from different "RaiseArm" and "Wave" captures.

As expected, when comparing vectors from different gestures, the distance obtained is much higher than before. In this case, almost double the previous comparison. Table 5.3 shows the distances between repetition 1 of "Raise Arm" and all repetitions of "Back and Forth".

Repetitions Compared	Euc. Distance
1-1	1830
1-2	576
1-3	197
1-4	1237
1-5	1144
1-6	616
1-7	943
1-8	263
1-9	1981
1-10	725
1-11	812
1-12	1484
1-13	1682
1-14	847

Table 5.3: Distances obtained when comparing "RaiseArm1" with all repetitions of "Back And Forth"

The average distance was 1024.1 with a standard deviation of 559.69, a maximum value of 1981 and a minimum of 197. The next table show the distance between a few selected repetitions of one gesture compared to all repetitions of other gestures. Table 5.4 shows the distances between repetition 1 of "Raise Arm" and all repetitions of "Wave".

Repetitions Compared	Euc. Distance
1-1	408
1-2	294
1-3	265
1-4	581
1-5	299
1-6	430
1-7	199
1-8	269
1-9	426
1-10	366
1-11	349
1-12	351
1-13	744
1-14	286

Table 5.4: Distances obtained when comparing "RaiseArm1" with all repetitions of "Wave"

The average distance was 376.2143 with a standard deviation of 141.75, a maximum value of 744 and a minimum of 199. At first the results seemed promising as the comparison between different gestures yielded a high distance value. However, Table 5.4 yields much smaller distance values while comparing different gestures.

After obtaining these results, we realized that storing the data from the radar as vectors means losing some information relative to the patterns. So it was decided to transform the

data into images. The images were obtained using the methods described in section 4.5. Also, it was decided to use a different library to calculate DTW and obtain the cost matrix and warping curve, called tslearn. This change is due to the fact that tslearn allows for better visualization of the results.

The following images show the plots obtained by the tslearn library. The two figures 5.10 and 5.11 show the results obtained from the comparisons. Figure 5.10 shows the comparison between repetitions of the same gesture while 5.11 shows the comparison between different gestures. The top-right and top-left plots show the images compared, bottom-left shows the warping curve and the cost matrix. The warping curve is the white line which depicts the differences along the images. The closer this lines is to a diagonal, the smaller the distance between compared images.

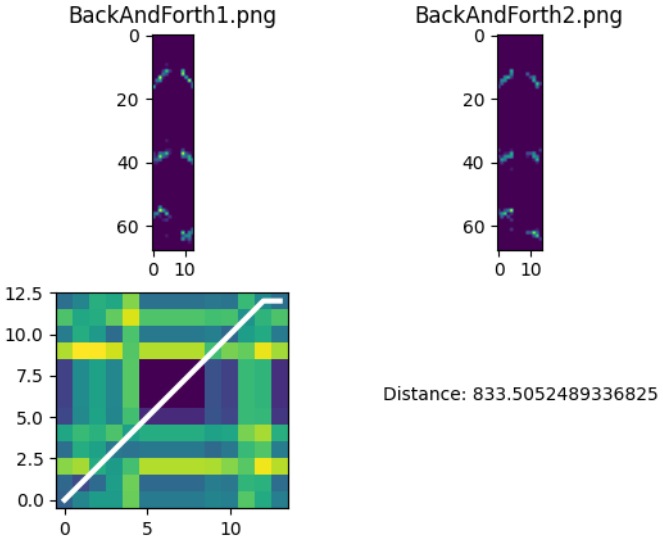


Figure 5.10: Visual representation of the cost matrix and warping curve (bottom-left) obtained with DTW, when comparing two repetitions of the "Back and forth" gesture. The associated distance is also indicated (bottom-right). The images corresponding to the two gesture repetitions are shown at the top of the figure.

Image 5.11 shows the comparison between repetitions of different gestures.

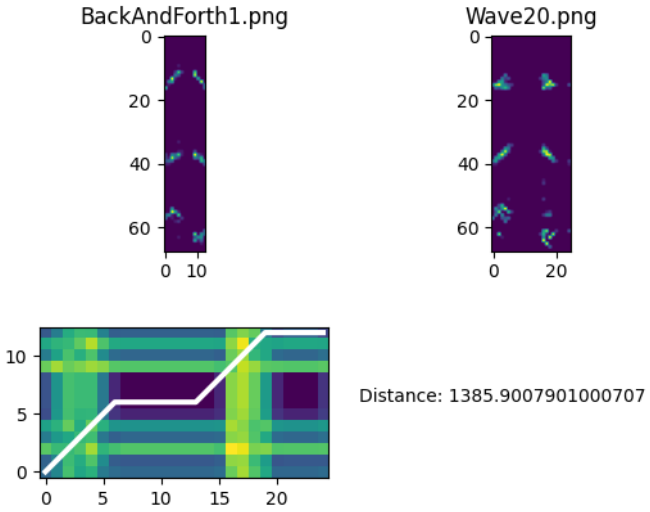


Figure 5.11: Visual representation of the cost matrix and warping curve (bottom-left) obtained with DTW, when comparing one repetition of the "Back and forth" and Wave gestures. The associated distance is also indicated (bottom-right). The images corresponding to the two gesture repetitions are shown at the top of the figure.

As expected, the comparison between different gestures yields a higher distance and a less diagonal warping curve. This conclusions go according to what was expected, and show us that there is enough differences in the patterns of the gestures to differentiate them. It also gives us confidence that using images for classification was the right approach to take.

5.3 Initial Evaluation with a Single Subject

Due to the good results obtained from using DTW with images, we decided to explore the possibility of using transfer learning to see whether pre-trained deep learning models for image recognition are able to adapt to their capabilities to the gesture images. An initial evaluation was performed with a single Subject to explore that possibility.

5.3.1 Experimental Setup and Protocol

Radar data were captured from a 23-year-old, right-handed male Subject. The used setup is the one included in Fig. 4.1, where the Subject was lying on the bed on their back. Three different gestures, "BackAndForth", Wave and RaiseArm, were executed 50 times each. Even though the Subject is right-handed, all gestures were performed with the left arm, due to the radar being on the left side of the bed. For each repetition, data recording was initiated before the gesture execution and stopped automatically after 5 seconds.

5.3.2 Dataset

For each capture corresponding to a gesture repetition, an image was generated as explained in section 4.5 The obtained dataset includes 150 images (50 per gesture). Since deep learning

requires a large dataset to obtain reasonable results, we expanded the dataset relying on offline dataset augmentation, which is explained in section 4.6. In this case, two different dataset were obtained from the original dataset: dataset augmented 1 and 2. Both datasets were generated using the classic noises approach.

5.3.3 Model Evaluation

To obtain a model that recognises the considered gestures, we used the transfer learning method. Since our aim is to run gesture recognition in a processing unit with limited memory and computing capability, from the pre-trained models directly available in Keras [46], we explored three that achieved a top-5 accuracy equal or greater than 90% (on ImageNet validation dataset) and have less than 10 million parameters: MobileNetV2, NASNetMobile, DenseNet121.

For each pre-trained model, the top layers only were replaced by a single fully connected layer with 256 neurons (using ReLU as its activation function) and an output layer with 3 neurons (softmax activation function). The used optimizer was ADAM (default parameters). Crossentropy was used as the loss function, and accuracy as the metric to be evaluated during training and validation.

Each model was evaluated using a variation of the the 10-fold cross-validation approach, where 80% of the dataset is used for training, 10% for validation, and 10% for testing, in each iteration. Cross-validation splits the data into mini train-test sets allowing the test sets to be data unseen. Training is stopped when the validation loss has not decreased more than 0.1 for 5 epochs. This is called early stopping. Early stopping is a way to stop a learning iteration based on a condition. The technique stops the iteration at a point where the models ability to generalize can not improve. The resulting model is evaluated on the test data of the corresponding iteration.

The results presented in 5.3.4 were obtained with Google Colaboratory, a service that allows the user to write Python code in their web app and have access to machines to execute the said code. In this case, an Intel(R)Xeon(R) CPU running at 2.30 GHz provided the results. As Google Colaboratory shares the machines with all users, the service allocated only one core to our testing.

As there is no way to guarantee that the machine allocated by Google Colaboratory is always the same, we ran the evaluations with multiple Subjects locally (section 5.4). We obtained the results with a computer with an AMD Ryzen 5 5600X with six cores, twelve threads running at 3.7GHz.

5.3.4 Results and Discussion

Results were obtained for the three pre-trained models listed above and for three different datasets: original (150 images); augmented 1 (750 images); augmented 2 (1500 images). The boxplots for the accuracy, F1 score, train time, and prediction time per image, considering all cross-validation folds, are shown in Fig. 5.12.

Examining the boxplots present in Fig. 5.12 allows for several observations. The graphs show the effects of different pre-trained models and types of augmentation. Increasing the

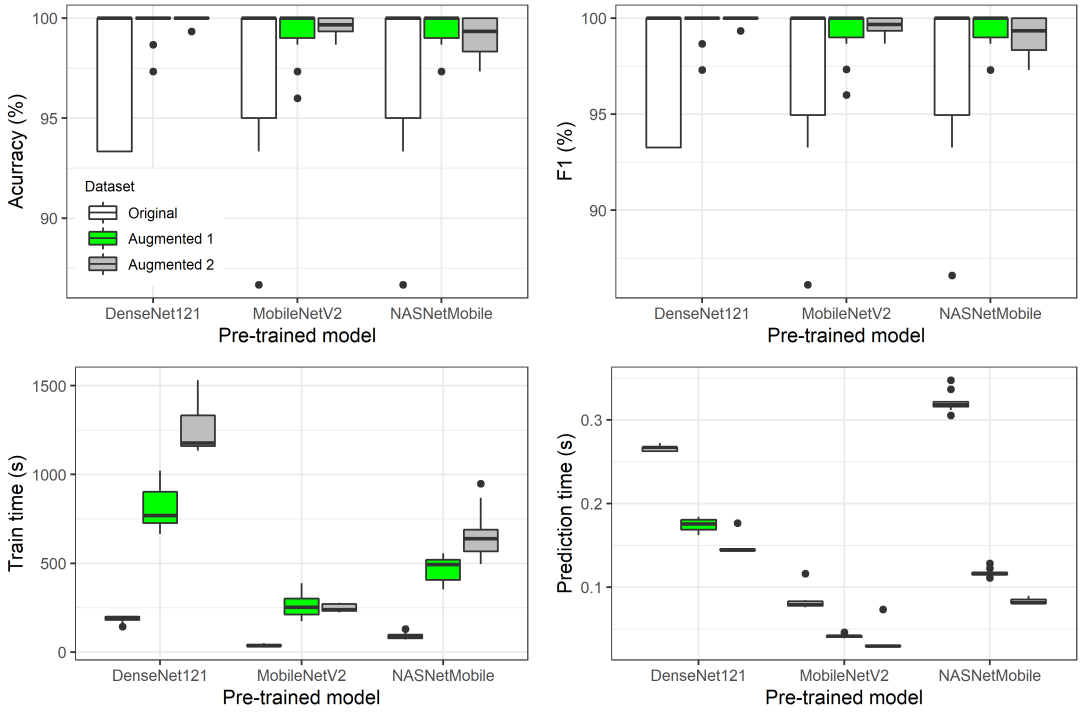


Figure 5.12: Boxplots for the accuracy (left-top), F1 score (right-top), train time (left-bottom), and prediction time (right-bottom), for each model and dataset.

size of the dataset has a positive outcome on the accuracy and F1 score. This was expected, as the model has more data to use in the training process. Although augmentation increases performance, the train times increase when the size of the dataset increases, as expected.

Regarding accuracy and F1 score values, the best combination present is DenseNet121 and augmented dataset 2. This model however does have the most variability in accuracy and F1 score when using the original dataset. Although the model shows the best values in terms of accuracy and F1, ti is also the slowest model, with high training and prediction times. MobileNetV2 and NASNetMobile show very similar results between each other with the main difference being for augmented dataset 2 where MobileNetV2 has a slightly higher accuracy and F1 score. When regarding training and prediction times, the difference among datasets is lower for the MobileNetV2 model, which also has the lowest median train and prediction times: 35 to 252 s and 0.03 to 0.08 s, respectively, versus 84 to 639 s and 0.08 to 0.32 s for NASNetMobile, and 185 to 1177 s and 0.14 to 0.27 s for DenseNet121. This was also expected, since MobileNetV2 is the smallest of the three pre-trained models ($\approx$3.5 M parameters), followed by NASNetMobile ($\approx$5.3 M parameters; DenseNet21 has $\approx$8.1 M). Despite of its smaller size, MobileNetV2 still leads to a model with a median accuracy and F1 score similar to the other models ($\geq$99% for all datasets). This model is a convolutional neural network architecture that contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers [47]. Although these results are quite good, it can be because only three gestures were considered and all used data came from the same Subject.

With the aim of verifying the previous results, and exploring the possibility of having a user independent solution, a new experiment was carried out with data acquired from more than one Subject.

5.4.1 Participants and Gestures

For this evaluation, we obtained data from 4 different Subjects (two male and two female) with an ages ranging from 23 to 33 and an average of 26. All four Subjects used their right arm to perform the gestures. We decided to added two more gestures compared with the previous evaluation with a single Subject (i.e., 5 gestures were considered). A description of all captured movements was presented previously in table 4.1.

Before each recording session, the participants signed a declaration of consent. This declaration described the objectives of the experiment, the procedure, among others. All Subjects had an unique identifier assigned to guarantee anonymity and confidentiality.

5.4.2 Experimental Protocol

The used setup was the same as the one used in the previous evaluation (section 5.3.1). Regarding the protocol, all Subjects had access to a table describing the gestures and were encouraged to execute the movement freely, varying speed and amplitude.

During the sessions presented in the previous section (section 5.3), many captures ended up being unusable due to some of the data being corrupted. This meant repeating the captures, which took a lot of time. It was decided to change the capture procedure to save time. To check whether the data are corrupted or not, we created a Python script that reads the packet header while the recording session is still in progress. This verification allows the Subject to stop executing the gesture repetitions, if needed, and start a new capture without losing too much time. The Subjects would execute ten repetitions of the same gesture in each capture. The Subjects were also instructed to count to five between each repetition in order to have an interval between repetitions. The repetitions are segmented and extracted.

5.4.3 Dataset and Evaluation Method

As explained above, each capture includes ten repetitions of a given gesture. For each gesture, 2 captures were performed. Therefore, 10 captures were carried out with each Subject, which corresponds to 40 captures in total. For capture, the different repetitions were segmented based on the absence of movement during intervals around five seconds.

To have a balanced dataset in terms of both class (gesture) and group (Subject), the same number of repetitions per gesture were selected randomly for each Subject. In this case, ten repetitions were chosen per gesture and Subject, resulting in a dataset with two hundred examples or images.

To evaluate the influence of data augmentation and the performance of the proof of concept regarding the Subjects, the types of evaluation are divided into Subject dependent, Subject independent in which several approaches to data augmentation are used.

5.4.4 Subject Dependent

Two different evaluations were made for this section:

- **Subject dependent 1** (Sub.Dep.1): Train and test a model for each Subject separately;
- **Subject dependent 2** (Sub.Dep.2): Train and test a model using data from all Subjects (for both training and testing);

5.4.5 Subject Independent

Two different evaluations were made for this section:

- **Subject independent 1** (Sub.Indep.1): Train a model using data from all Subjects except one and test it with data from the remaining Subject, and repeat the process for all Subjects;
- **Subject independent 2** (Sub.Indep.2): Train a model using data from one Subject and test it with data from the remaining Subjects, repeating the process for all Subjects;

5.4.6 Data augmentation

For all evaluations, six different datasets were obtained from the original dataset by using different types of offline augmentation. The types of augmentation were:

- **Shift augmentation**: Images are augmented by performing a horizontal shift (the amount is chosen randomly between (-0.5 and 0.5) pixels over the original image, which simulates a sliding window;
- **Noise augmentation 1**: Images are augmented by adding classic noise(s) to the original image as explained in 4.6;
- **Noise augmentation 2**: Images are augmented by adding noise blocks to the original image as explained in 4.6.

The datasets are the following:

- **Dataset without augmentation** (200 images) - Original dataset, without any augmentation;
- **Dataset augmented 1** (1000 images) - Dataset resulting from shift augmentation over the whole original dataset (5 new images per original image);
- **Dataset augmented 2** (5040 images) - Dataset resulting from shift augmentation over the whole original dataset (5 new images per original image) and then noise augmentation 1 over the training set only (5 new images per augmented image);
- **Dataset augmented 3** (5040 images) - Dataset resulting from shift augmentation over the whole original dataset (5 new images per original image) and then noise augmentation 2 over the training set only (5 new images per augmented image);
- **Dataset augmented 4** (9840 images) - Dataset resulting from shift augmentation over the whole original dataset (5 new images per original image) and then noise augmentation 1 over the training set only (10 new images per augmented image);
- **Dataset augmented 5** (9840 images) - Dataset resulting from shift augmentation over the whole original dataset (5 new images per original image) and then noise augmentation 2 over the training set only (10 new images per augmented image);

According to the results of section 5.3, the pre-trained model used in all these evaluations was the MobileNetV2.

5.4.7 Subject Dependent 1 Results

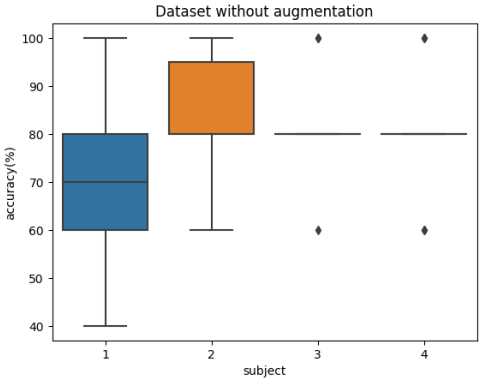


Figure 5.13: Sub.Dep.1: Boxplot for the accuracy obtained for each Subject, when using the dataset without augmentation

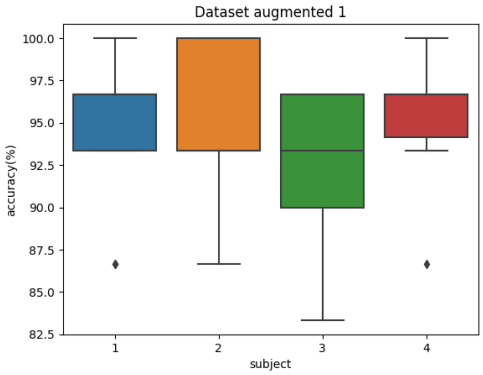


Figure 5.14: Sub.Dep.1: Boxplot for the accuracy obtained for each Subject, when using the dataset augmented 1

Figure 5.13 shows the boxplot for the accuracy values obtained for each Subject, when using the dataset without any augmentation. This boxplot shows us that the accuracy values from one Subject to another do not vary much. Subjects 2, 3 and 4 all have a median accuracy of around 80%, while dataset one has 70%. Subject 1 also has a higher variability due to presenting a high maximum value and a low minimum value. One possible explanation for these values is a higher difference in gesture execution for Subject 1 comparing with the other Subjects. As the recording session progressed, this Subject may have slightly changed the way one or more gestures were carried out movement for a gesture, making the different repetitions for the same gesture different among them.

Figure 5.14 shows the accuracy values for the same Subjects, but this time, the used data corresponds to dataset augmented 1. Instantly, we can see the benefits of data augmentation. The median accuracy is elevated, with the lowest value increasing to around 93%. Once again, Subject 2 reaches an accuracy around 100%. Besides the median, Subject 3 shows a higher variability with a minimum value around 83%.

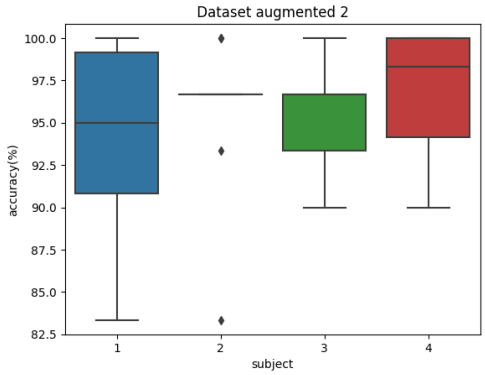


Figure 5.15: Sub.Dep.1: Boxplot for the accuracy obtained for each Subject, when using the dataset augmented 2

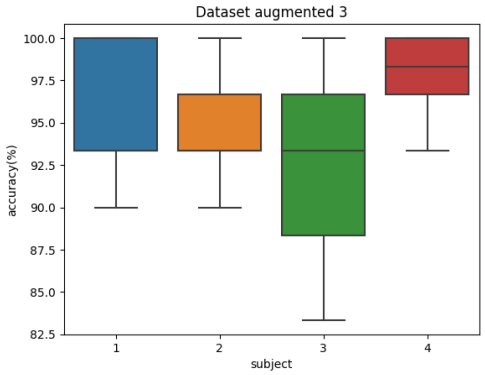


Figure 5.16: Sub.Dep.1: Boxplot for the accuracy obtained for each Subject, when using the dataset augmented 3

Both Fig. 5.15 and 5.16 show the accuracy values for each Subject, with Fig. 5.15 corresponding to dataset augmented 2 and Fig. 5.16 to dataset augmented 3.

Augmenting the training set with classic noise or noise blocks yielded positive results, with all Subjects of figure 5.15 showing a median accuracy equal or above 95%. In figure 5.16, all Subjects had a median accuracy above 93%. It is also possible to see that the variability in both boxplots is similar, with Subject 1 from 5.15 and Subject 3of 5.16 having the same minimum of around 83% and maximum of 100%. It is also to note that Subject 2from 5.15 shows very little variability.

When using the dataset augmented 4 and dataset augmented 5,no considerable changes to the median, minimum or maximum accuracy values have appeared.

Figures 5.17 to 5.22 show the same results as figures 5.13 to 5.16 when considering the F1 score. After analysing the plots, we can see that the F1 score values are similar to the accuracy from before.

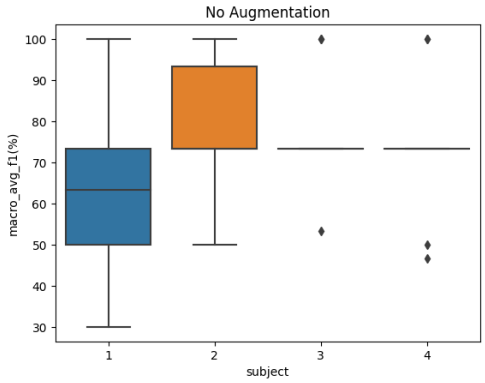


Figure 5.17: Sub.Dep.1: Boxplot for the F1 score obtained for each Subject, when using the dataset without augmentation

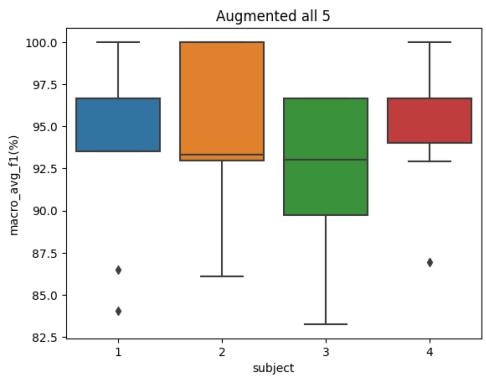


Figure 5.18: Sub.Dep.1: Boxplot for the F1 score obtained for each Subject, when using the dataset augmented 1

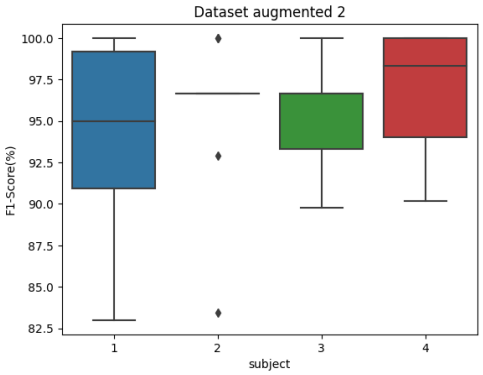


Figure 5.19: Sub.Dep.1: Boxplot for the F1 score obtained for each Subject, when using the dataset augmented 2

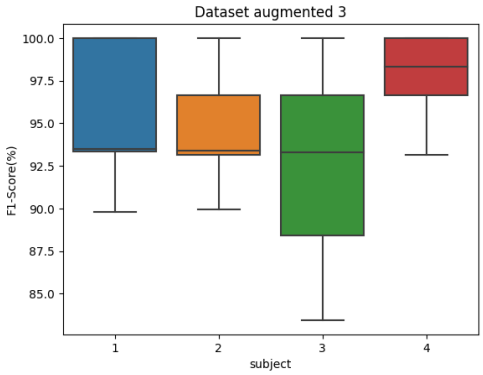


Figure 5.20: Sub.Dep.1: Boxplot for the F1 score obtained for each Subject, when using the dataset augmented 3

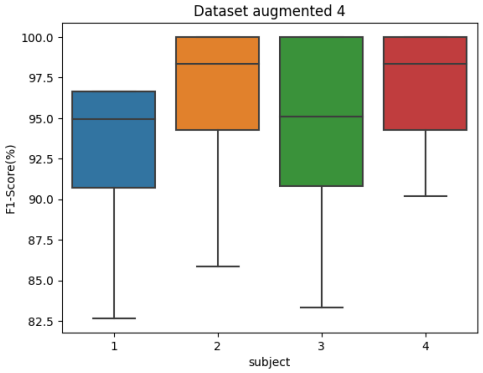


Figure 5.21: Sub.Dep.1: Boxplot for the F1 score obtained for each Subject, when using the dataset augmented 4

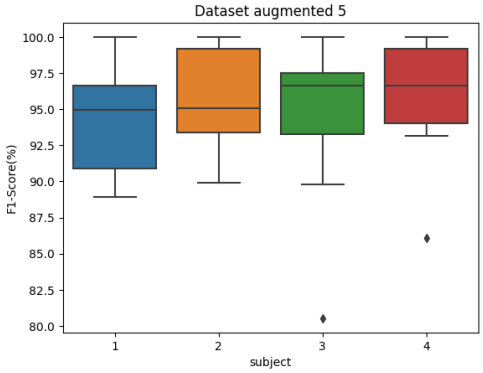


Figure 5.22: Sub.Dep.1: Boxplot for the F1 score obtained for each Subject, when using the dataset augmented 5

The following boxplots show the different values for train and predict times as the type of augmentation changes. Figures 5.23 and 5.24 illustrate the train and predict times, respectively, for the non-augmented dataset. As the dataset is relatively small, both operations are quick. The median for training time is around six seconds long except for with Subject 3 having a slightly higher median of eight seconds. As for the prediction time per image, all datasets have medians around 68 to 70 milliseconds.

As we increase the size of the dataset, training time increase. This is to expected as there are mode data to train. In Fig. 5.23 we see median times ranging from 5 to 8 seconds. I Fig. 5.25, we see this interval rise to 24 to 26 seconds. The prediction times per image although, decrease in Fig. 5.26 when compare to Fig. 5.24. All median values are within 18 to 19 milliseconds. As the training set in larger, the model has more data to train with, leading to faster prediction times per image. Fig. 5.25 shows an increase in interquartile range and in variability when compared to 5.23.

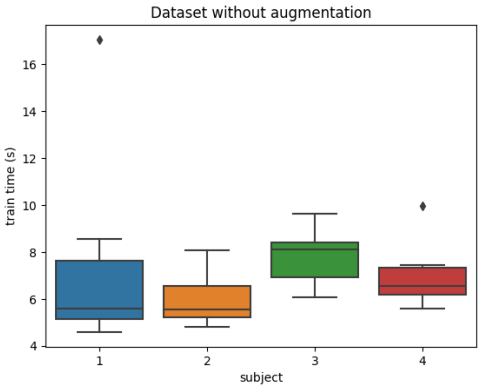


Figure 5.23: Sub.Dep.1: Boxplot for the model train time for each Subject, when using the dataset without augmentation

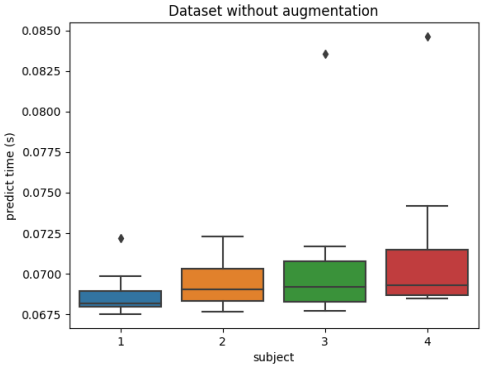


Figure 5.24: Sub.Dep.1: Boxplot for the model prediction time per image for each Subject, when using the dataset without augmentation

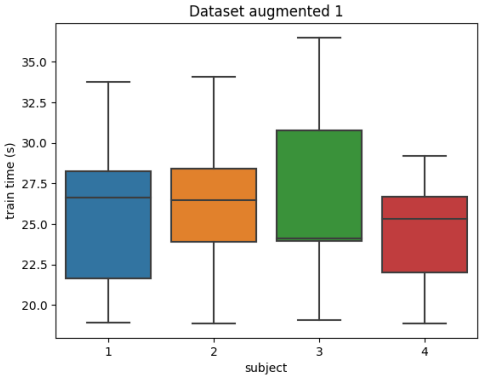


Figure 5.25: Sub.Dep.1: Boxplot for the model train time for each Subject, when using the dataset augmented 1

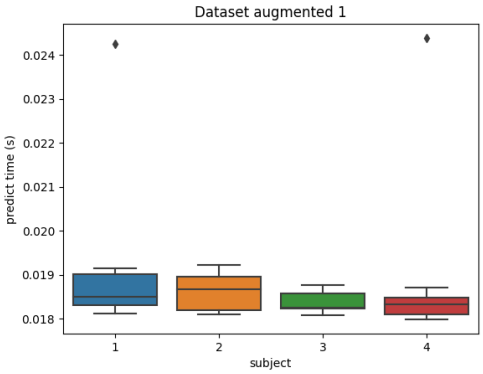


Figure 5.26: Sub.Dep.1: Boxplot for the model prediction time per image for each Subject, when using the dataset augmented 1

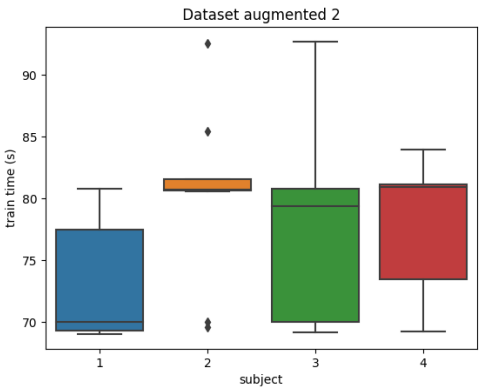


Figure 5.27: Sub.Dep.1: Boxplot for the model train time for each Subject, when using the dataset augmented 2

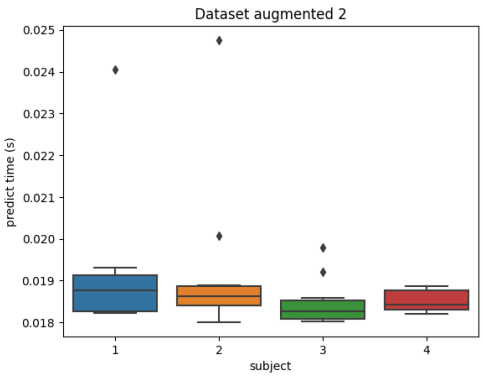


Figure 5.28: Sub.Dep.1: Boxplot for the model prediction time per image for each Subject, when using the dataset augmented 2

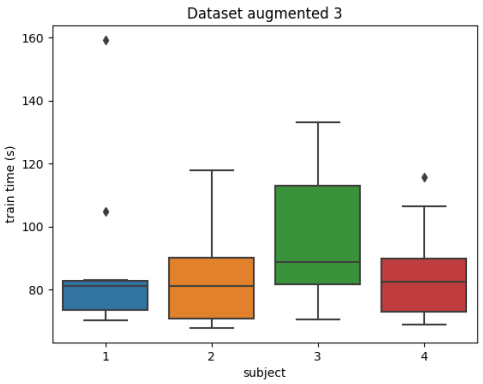


Figure 5.29: Sub.Dep.1: Boxplot for the model train time for each Subject, when using the dataset augmented 3

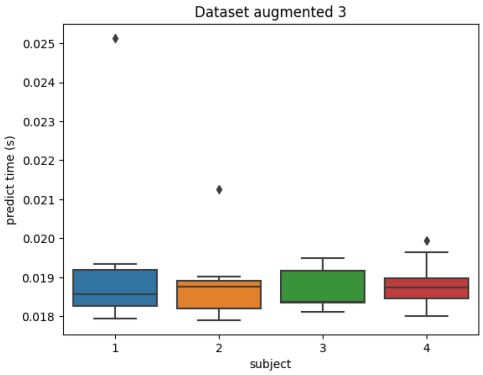


Figure 5.30: Sub.Dep.1: Boxplot for the model prediction time per image for each Subject, when using the dataset augmented 3

Figures 5.27 to 5.30 show the training and prediction times for augmented dataset 2 to augmented dataset 3. Figures 5.27 and 5.28 had the training set augmented with Noise augmentation 1, while the two latter figures used Noise augmentation 2.

When comparing the training times from augmented dataset 2 (Fig. 5.27) and 3 (Fig. 5.29) we see that the values are very similar. In Fig. 5.27 the median values range from 70 to 83 seconds. In Fig. 5.29 the median values range from 80 to 90 seconds. As expected, when we compare these datasets to augmented dataset 1 and the dataset without augmentation, there is a considerable increase in training times. However, this is not the case for prediction times, as shown in Figs. 5.28 and 5.30. These values are very similar to the ones presented in Fig. 5.26.

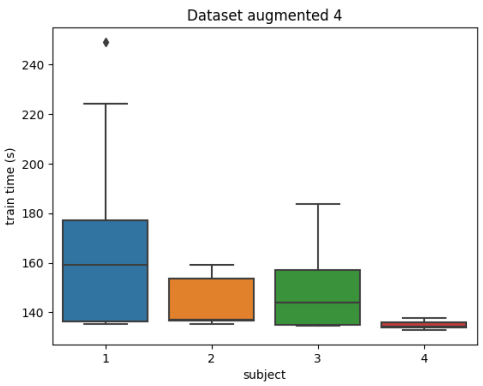


Figure 5.31: Sub.Dep.1: Boxplot for the model train time for each Subject, when using the dataset augmented 4

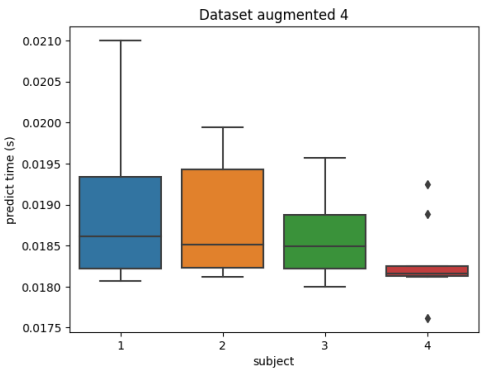


Figure 5.32: Sub.Dep.1: Boxplot for the model prediction time per image for each Subject, when using the dataset augmented 4

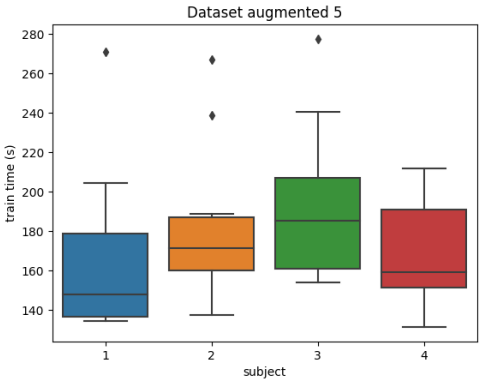


Figure 5.33: Sub.Dep.1: Boxplot for the model train time for each Subject, when using the dataset augmented 5

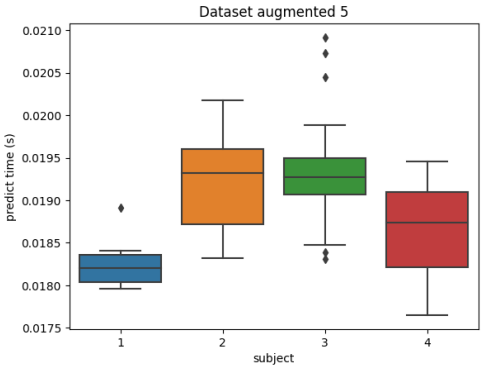


Figure 5.34: Sub.Dep.1: Boxplot for the model prediction time per image for each Subject, when using the dataset augmented 5

Increasing the the number of new images generated per image from 5 to 10 when augmenting the training set yields an increase in training times and similar results for prediction times. When we compare Fig. 5.33 to Fig. 5.29 the increase in training times is clear to see. The median values increase from a range of 80 to 90 seconds to an interval of around 130 to 160 seconds. Fig. 5.31 shows an even higher interval with values ranging from 145 to 190 seconds.

Regarding prediction times, once again the median values range from 18 to 19 milliseconds.

5.4.8 Subject Dependent 2 Results

As mentioned in chapter 5.4.3, this evalution uses data from all Subjects.

For this evaluation, Fig. 5.35, 5.36, 5.37, 5.38 show the boxplot for the accuracy, F1 score, training time, and prediction time per image, respectively, obtained for each considered dataset.

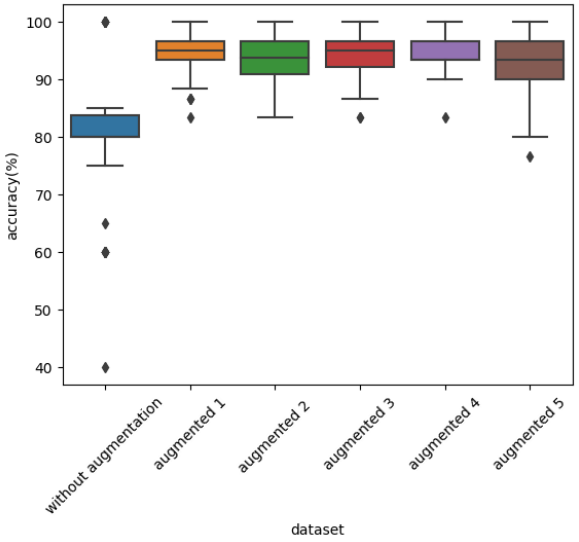


Figure 5.35: Sub.Dep.2: Boxplot for the accuracy values obtained for each dataset.

Looking at Fig. 5.35, it is clear to see, once again, the benefits of data augmentation.

The median value for each augmentation dataset is similar, as all augmented datasets are within 93 to 100%, with a slight decrease for the dataset without augmentation whose median value is around 80 to 85%. These results match with the ones from section 5.3, as once again the augmented datasets have similar accuracy regardless of the number of generated images. This may be due to the fact that the augmented images are slightly modified versions of the originals, meaning that the variability of the dataset increases slightly at best. As the increase in variability is very limited, the model is able to get similar accuracy from both types of dataset.

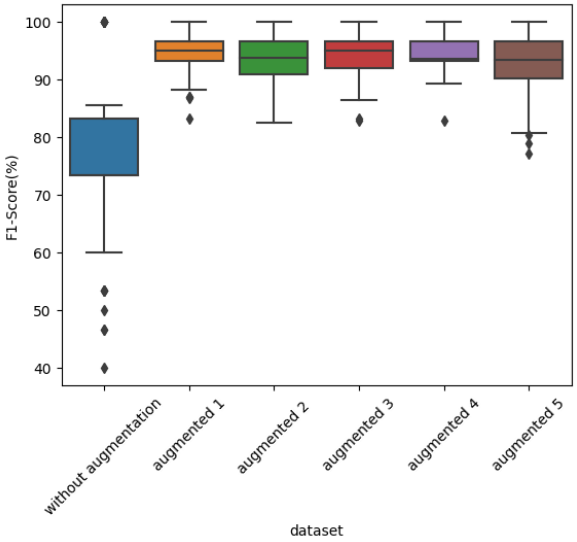


Figure 5.36: Sub.Dep.2: Model f1-score for each degree of augmentation

The F1 score values are similar to the accuracy values.

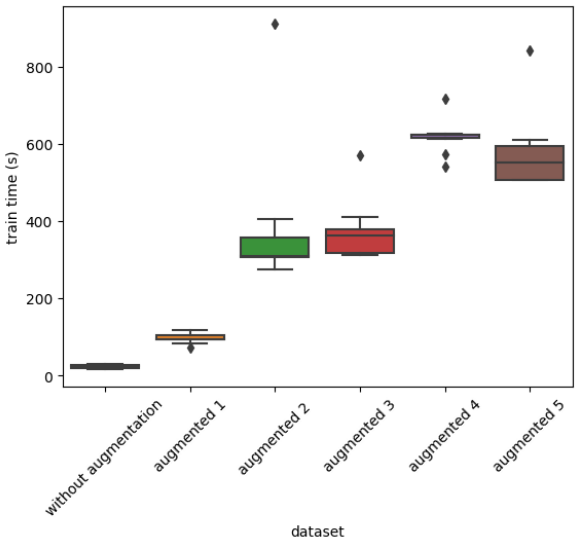


Figure 5.37: Sub.Dep.2: Model training times for each degree of augmentation

As the dataset contains the data for all Subjects, we anticipated an increase in training times. In test 1, section 5.3, the maximum value for the training time was around 250 seconds when using dataset 4 and 5 which are the datasets that include the highest number of images. Due to a higher dataset size, we see training times above 200 seconds with augmented dataset 3, with this dataset including an outlier above 900 seconds. Besides this, the same conclusions are visible as before, i.e., the higher the dataset size, the higher the training times, which is according to the initial expectations.

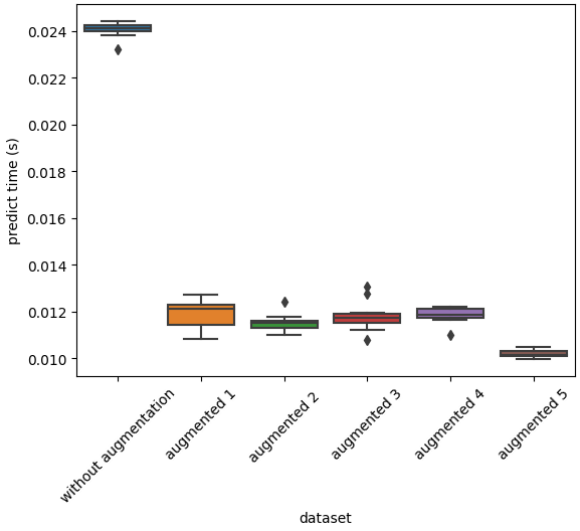


Figure 5.38: Sub.Dep.2: Model prediction times for each degree of augmentation

Fig. 5.38 allows for the same conclusions for prediction times as the first Subject-dependent evaluation. The dataset with no augmentation shows a prediction time per image of 25 milliseconds. These values are halved as soon as an augmented dataset is used. All other datasets have prediction times ranging from 10 to 13 milliseconds.

After analysing the accuracy and f1-score values, we came to the conclusion that training a model with each Subject individually and all Subjects yield similar results. This may be due to the fact that the same Subject executed all gestures, meaning that all repetitions will be similar, mitigating the effect of a smaller dataset. Training all Subjects together allows for a much larger dataset, but will also have slightly different executions for the gestures which can impact performance.

As expected, training each Subject individually allows for shorter training times.

5.4.9 Subject Independent 1 Results

The accuracy, F1 score, training and prediction time results obtained for each considered dataset, for the first Subject-independent evaluation, are shown in Fig. 5.39, 5.40, 5.41, 5.42, respectively.

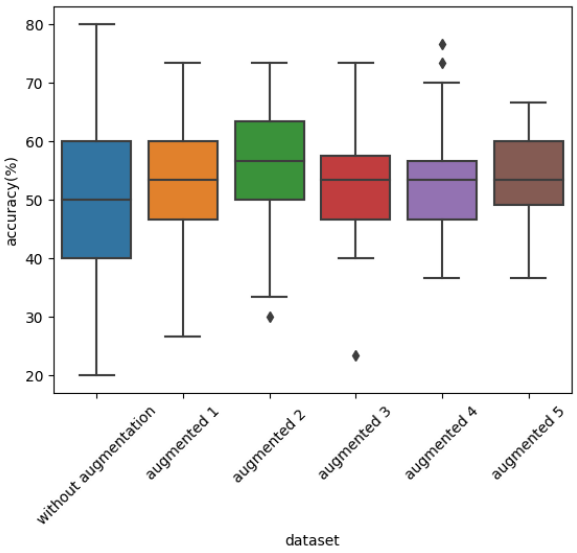


Figure 5.39: Sub.Indep.1: Model accuracy values for each degree of augmentation

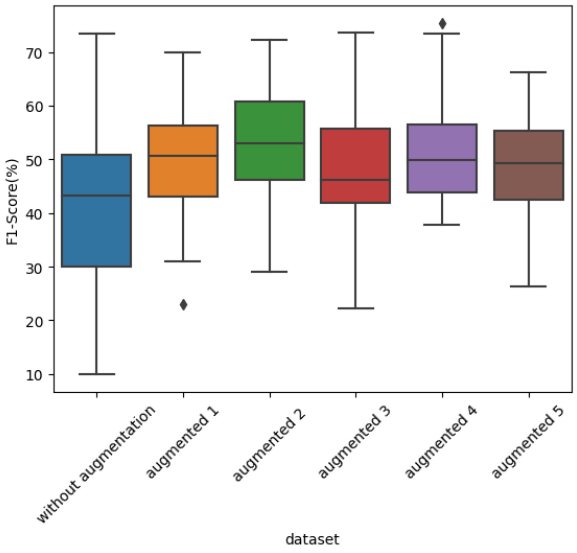


Figure 5.40: Sub.Indep.1: Model f1-score for each degree of augmentation

Looking at Fig. 5.39, we see that the median accuracy values do not increase a lot when dataset augmentation is performed, as it happened in the Subject depedent evaluations. All median values range from 50 to 55%. Nevertheless, the increase in the number of images due to augmentation does have a positive effect. When comparing the values for the dataset without augmentation to the augmented ones, it is clear that the interquartile range is smaller, meaning there is less variability when more data are used.

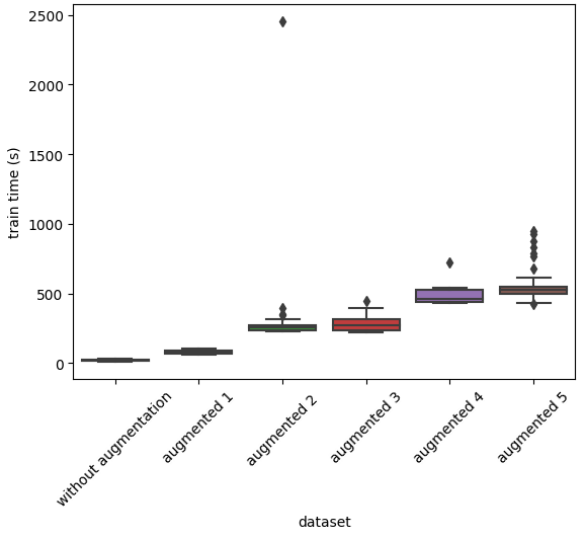


Figure 5.41: Sub.Indep.1: Model training times for each degree of augmentation

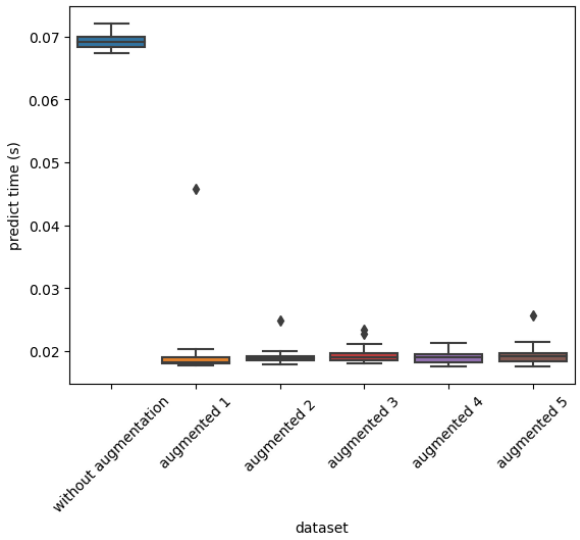


Figure 5.42: Sub.Indep.1: Model prediction times for each degree of augmentation

Figure 5.40 allows for the same observation as Fig. 5.39. The increase in dataset size leads to a decrease in interquartile range when comparing to a dataset with no augmentation. Once again, the median values do not vary much, from 45 to 55%.

Regarding training times, these are not very different from the ones in the second Subject-dependent evaluation, since the number of Subjects used for training is similar. Predictably, the datasets with higher augmentation had longer train times. It is interesting to note that there is an increase in outliers, including one close to 2,500 seconds.

Prediction times show values that almost double the ones in the previous evaluation for augmented datasets. The biggest increase is in the no augmentation dataset that rises from 24 to 70 milliseconds. These results are expected, as they are very similar to the ones obtained in evaluation one, section 5.4.7, as the dataset used for training has the same size in both tests.

5.4.10 Subject Independent 2 Results

The accuracy, F1 score, training and prediction time results obtained for each considered dataset, for the first Subject-independent evaluation, are shown in Fig. 5.43, 5.44, 5.45, 5.46, respectively.

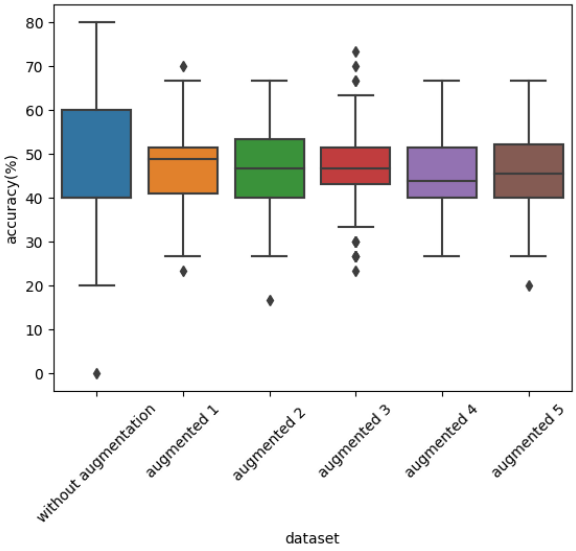


Figure 5.43: Sub.Indep.2: Model accuracy values for each degree of augmentation

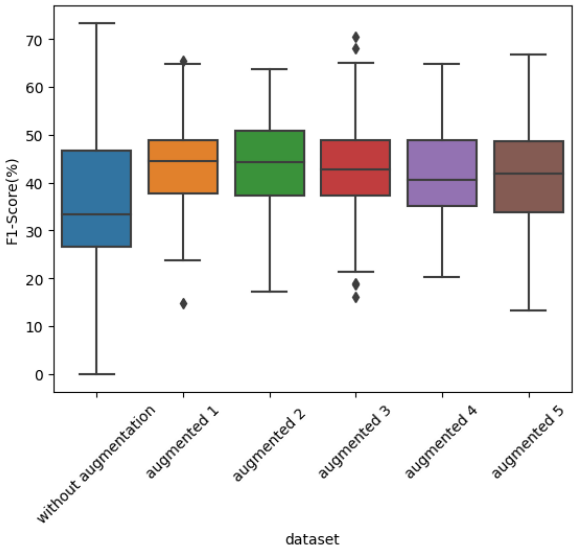


Figure 5.44: Sub.Indep.2: Model f1-score for each degree of augmentation

Looking at Fig. 5.43, once again, we see that augmentation increases the median results, from 40% for dataset without augmentation and between 45 and 50% for the augmented datasets. It does also, have an impact on the variability of the results, as the dataset with no augmentation has a much higher maximum value and a lower minimum than the others.

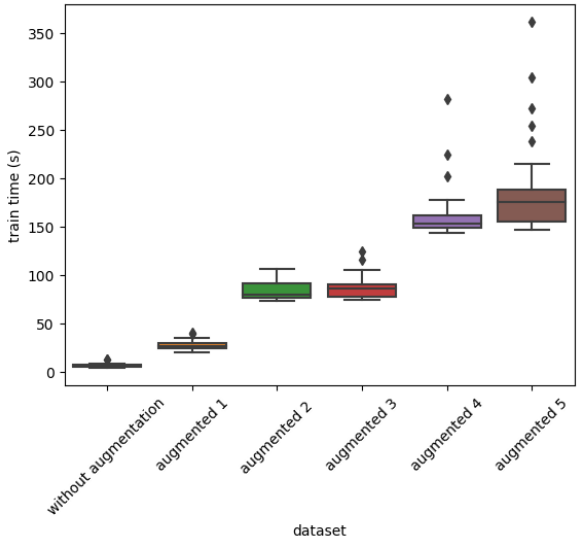


Figure 5.45: Sub.Indep.2: Model training times for each degree of augmentation

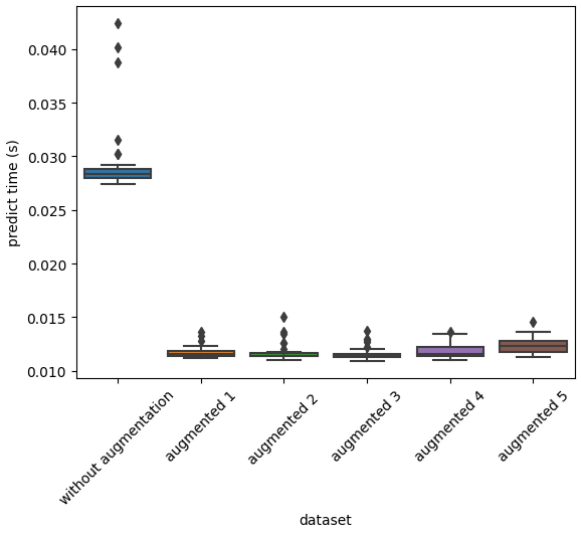


Figure 5.46: Sub.Indep.2: Model prediction times for each degree of augmentation

The F1 score backs up the conclusion exposed in the last paragraph, as Fig. 5.44 shows similar factors. Train times suffered a considerable decrease relative to the results from the previous test, which is explained by the size of the training set. As in this evaluation, the model uses only data from one Subject for training, this phase is shorter.

For the prediction times, not much has changed. Once again, the dataset with no augmentation has the highest prediction time, while the augmented datasets show similar median values.

Is is also important to note that the accuracy results are lower than the ones from the evaluations before. This may be explained by the fact that a dataset from a single Subject does not contain enough variability within itself to be able to properly train a model to recognize never before seen data from other Subjects.

5.5 Conclusion

This chapter describes the development stages for the prototype of the system. These include the initial exploration of the radar and its capabilities, the first data captures, exploration of the data captured, and finally classification with a single and later with multiple Subjects.

The first experiments aimed to further our understanding of the radar and the data it captured. After performing the first captures, the data was processed to try and find patterns or features. These then helped us decide what techniques to use for gesture recognition. It was decided to use transfer learning to train a model and use images created from the radar data as input.

The first evaluation used only data from a single Subject. This evaluation allowed us to test several models and eventually choose the most adequate, which was MobileNetV2. This model was also used in the following evaluations, which now involved data from multiple Subjects. These evaluations considered Subject-dependent and Subject-independent solutions. The results obtained allow us to conclude that the Subject-dependent solution allows for higher accuracy and F1 score with the highest median accuracy around 99%. The highest median accuracy value for a Subject-independent solution was around 58%.

These evaluations also allowed us to come to conclusions regarding data augmentation. In all cases, data augmentation has a positive effect on performance, except when considering training time which is to be expected. However, the increase in performance stagnates after a certain dataset size, so a moderate approach to the data augmentation technique used is the most appropriate.

CHAPTER 6

Conclusion

6.1 Work summary

This work consisted in gathering and reviewing literature regarding several important topics such as communication disorders, aphasia, gesture recognition, radar and developing a proof of concept of a radar-based gesture recognition system. With this in mind, the objectives of this dissertation have been met.

The development of dissertation went through several stages. As the work calls for an assistive system for people with communication difficulties, the first stage was to review literature regarding communication disorders and aphasia. This stage is essential as it helped us understand the target users and the problems that need addressing. Understanding the challenges the users face helps guarantee the system evolves towards answering these problems and provides support in the right areas. Besides this, it allows us to make choices that are better suited to the target users. For example, after reading the literature, realizing that strokes are one of the primary causes of aphasia, it became apparent that the gestures need to be easy to execute as stroke survivors can have motor limitations.

The next stage also revolved around reviewing literature, which helped understand how FMCW radar's function and the data it provides.

After understanding the basic relevant theory behind radar, it was time to start exploring the technology and its capabilities. This stage took quite some time as it was not easy to get started. After acquiring data, we created methods to process it. These methods allowed for a better understanding of these data and the possibilities for them. The data processing underwent various changes until eventually it was decided to transform the data into images to use as input for the model.

Now it was time to create a pipeline for gesture recognition to use as a proof of concept. ´ We explored several approaches to the problem before settling with the final one. The data captured goes through a filter to remove unwanted noise. After this, they are transformed into an image per data type (X, Y coordinates and Doppler index). The images are combined

into one and the final image is used as input for the model. This model was trained with transfer learning and carries out gesture recognition.

The final stage consists in evaluating the prototype of the radar-based gesture recognition system from the previous step. First, a single subject was recorded repeating three gestures. These captures yielded favourable results, but to ensure the functioning and ability of the system, it was necessary to capture data from different subjects. Recording sessions were setup with volunteers to gather data using the prototype of the system. These recording sessions yielded three more datasets to use. With these data, we were now able to confirm the results obtained for a single subject and to investigate the performance regarding subject dependent and independent solutions.

6.2 Main results

This work yielded several results. As the early stages of developments consisted in gaining theoretical knowledge, one of the results was a review of the literature regarding topics such as communication disorders, aphasia, gesture recognition and radar technology.

This work also resulted in a proof of concept system for gesture recognition using radar with encouraging results in classification.

As mentioned in the contributions, the exploration of the radar's capabilities and initial evaluation results also allowed the writing of an article entitled "Radar-Based Gesture Recognition Towards Supporting Communication in Aphasia: The Bedroom Scenario", recalling the development of the prototype. The article has also been accepted for publication at the conference EAI MobiQuitous 2021, the International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services.

Regarding the evaluation with a single subject, the results obtained are encouraging. All explored datasets and pre-trained models presented a median accuracy of $\geq$99%. Although these values are good, the same subject performing all gestures limits variability in the dataset, which may skew the results. Another factor to consider is that the recording process involved only three gestures. Regarding augmentation, the results show that although the median accuracy is very similar in all cases, the rise in dataset size lowers the variability both in accuracy and F1 score with the downside of requiring a larger training duration.

Regarding the evaluations with multiple subjects, they show the system's weakness. Training the model with each subjects data yields high median accuracies, with the lowest being 70% for the dataset without augmentation. All augmented datasets had median accuracies above 90%. These results support the ones obtained for evaluations with a single subject. Although the results are positive so far, this changes when we consider the results of the subject independent solution. This change lowers the median accuracy values to around 50 to 55%, which allows us to conclude that the prototype does not deal well with unseen data. In contrast, a subject dependent solution yields a relatively good performance: median accuracy of between 93 and 98% when performing dataset augmentation and considering an individual model for each user. Nonetheless, it is relevant to note that this type of solution

has implications when used in the real world since it requires data to be acquired for each new user.

6.3 Future Work

The work carried out established the grounds for exploring other approaches to better the results obtained and to expand in other directions.

Creating an online pipeline with the integration of the stages mentioned should be the priority for future work. With a few changes to the current code, and the addition of a single board of hardware, the DCA1000EVM from Texas Instruments, that connects to the radar,the pipeline can be updated from offline to online. This board allows real-time data capture and streaming. These changes, plus the addition of a smaller processing unit, for example a Raspberry Pi, would transform the prototype into a system much closer to a commercial solution.

Besides automating the pipeline and making the system run in real-time, adding more gestures to the set the system is able to recognize is also something to consider. The recognition of more gestures would allow the system to be more versatile and create more ways to assist the user. Another priority is also to gather more data from a greater number of subjects. Besides increasing the examples used to train, which can improve the results, it would also increase the variability of the gestures. The latter would allow the system to perform better with gesture data obtained from never-seen subjects.

Another feature to add to the system would be appropriate feedback to the user after the recognition of a gesture. This could be an audible cue such as sound, a simple sentence identifying the output of the gesture or even translating the meaning of the gesture into a speech generated sentence.

Finally, it is also possible to transform the skeleton of this prototype to work in different divisions besides a bedroom or even into a solution with a different purpose, such as a way to interact with appliances inside a house.

APPENDIX A

Radar Fundamentals

For this work, the radar selected is the AWR1642 from Texas Instruments. It is an integrated single-chip Frequency Modulated Continuous Wave (FMCW) radar sensor capable of operating in the 76 to 81 GHz band.

This model of radar, AWR1642 is a system with two transmitting (2TX) and four receiving (4RX) antennas and is equipped with an integrated C674x Digital Signal Processing (DSP) subsystem for radar Signal processing. The following image presents the axis of coordinates for the radar.

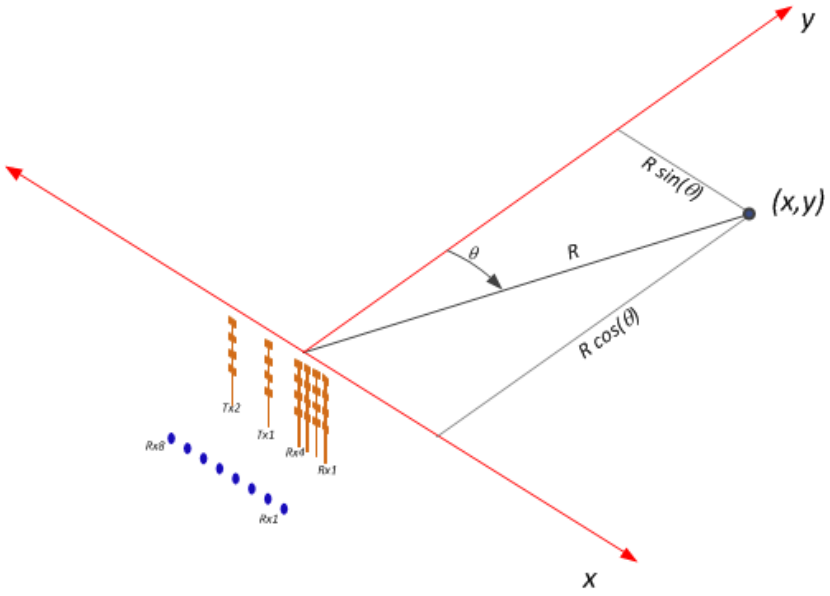


Figure A.1: Radar axis system. Taken from the Texas Instruments SDK doxygen

The most commonly used waveform for FMCW radars is the sawtooth with a linear increase in frequency. This type of signal is known as a chirp. In Fig. A.2, we can see a representation of the signal, where T_c represents the time interval, f_c represents the start

frequency, B bandwidth of the signal and the slope S, which characterizes the change in frequency [48].

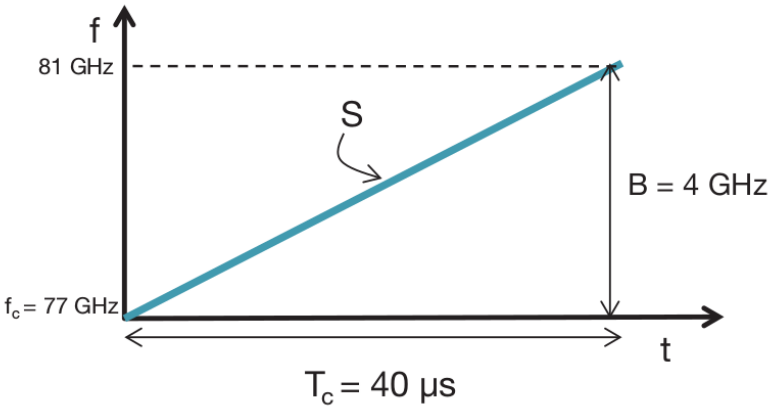


Figure A.2: Chirp signal, with frequency increase over time. Taken from [48].

A.1 Signal Generation

The radar contains a synthesizer module (module 1 in fig.A.3) which, as the name indicates, generate the chirps. Then, it uses the transmit (TX) antenna (module 2 in fig.A.3) to transmit the signal. These chirps will be reflected off objects in the way and received by the receive (RX) antenna (module 3 in fig.A.3). Both chirps are mixed (module 4 in fig.A.3). The resulting signal is called an Intermediate Frequency (IF) signal. This configuration is depicted in Fig. A.3.

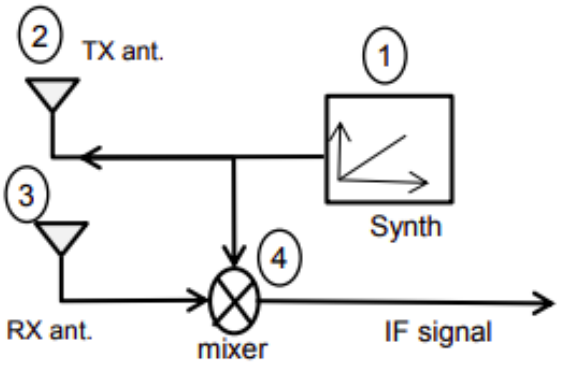


Figure A.3: Synthesizer, Transmission and Mixer modules. Taken from [48].

We can obtain the IF signal using the difference between the frequencies of the RX and TX signal. As the RX signal is a replica of the TX signal with a time delay, the IF signal obtained for a single object will be a constant frequency, as present in the images included in [48]. The process to obtain the IF signal is depicted in Fig. A.4.

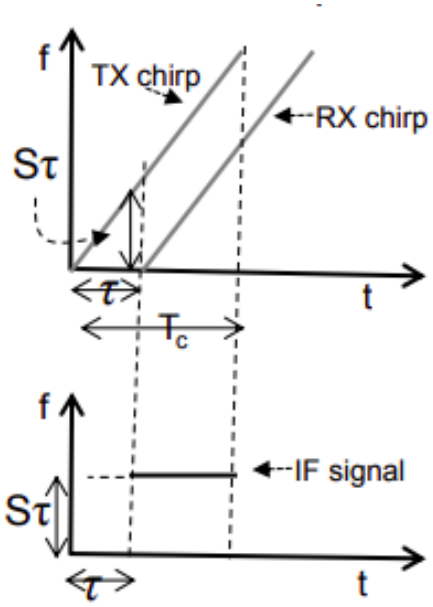


Figure A.4: RX, TX and IF signals. Taken from [48].

A.2 Fourier Transforms

The Fourier Transform is a mathematical transform that takes signals that are time or space-dependent and transforms them into frequency signals. As an example, a continuous sinusoidal wave produces only one peak in frequency, as can be seen in Fig. A.5.

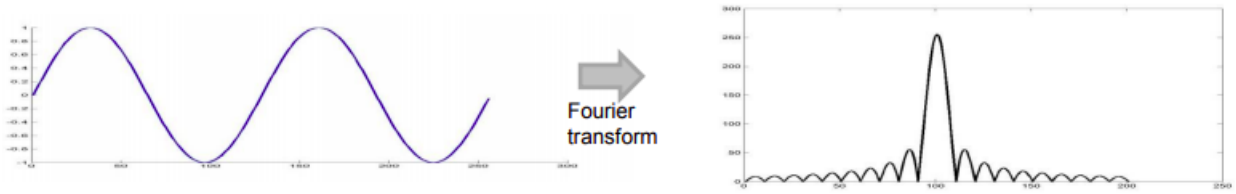


Figure A.5: Fourier Transform example. Taken from [48].

Applying this knowledge to a practical example, if the radar signal gets reflected by a target, then we can expect to see a peak in the frequency. Following this logic, if there are multiple targets in front of the radar there should be three peaks in the frequency. This behaviour is accurate if the targets are distant enough from each other, as similar distances will lead to similar IF frequencies [48].

A.3 Range Resolution

The concept of Range Resolution refers to the ability of the radar to distinguish targets and the minimum distance between them required to do so. A way to diminish the required minimum distance is to have a lengthier IF signal, as this allows for the waves of both targets to demonstrate their differences in frequency leading to two distinct peaks. The usual way to

increase the length of the IF signal is to increase the bandwidth of the chirp. This leads to the conclusion that the greater the bandwidth, the better the resolution [48], [49].

A.4 Signal Digitizing

After capturing data, these are converted from an analogue to a digital signal thanks to the Analogue-to-Digital Converter (ADC) module. This allows for further processing, for which the radar uses the Digital Signal Processing (DSP) module. This module produces the FFTs mentioned before [48], [49].

A.5 Velocity Measurement

For the radar to measure velocity, the adopted strategy is two transmit two chirps. The reflected chirps turn into IF signals and go through the range-FFT. According to the information present in section A.2, a single target will produce a single peak in the range-FFT. To extract velocity from a single peak in the range-FFT it is necessary to view this peak in a phasor notation [48], [49].

A.5.1 Phasor and Phasor Notation

A phasor, or phase vector, is a complex number that represents a sinusoidal wave, which in this case, contains the amplitude and phase of the sinusoid in each value [48], [49].

A.5.2 Single Target

After generating the range-FFT for both chirps, the peak value will be in the same frequency but with different wave phases. The velocity of the target represents the phase difference of the peak values in both range-FFTs [48], [49].

A.5.3 Multiple Targets

Unfortunately, the strategy applied before is only successful when trying to get the velocity for a single point. As mentioned previously, when multiple targets are equidistant from the radar, the range-FFT will return only a single peak. To get the velocity of various targets, the radar needs to transmit more than only two chirps. The adopted strategy uses a frame, which consists of a set of equally spaced chirps. These chirps are processed, the range-FFT returns a set of identically located peak values, each with the phases of the different targets. Finally, another FFT is applied, now to the peak values, called doppler-FFT. This FFT resolves the multiple targets and returns the velocity values [48], [49].

A.5.4 Velocity Resolution

The concept of Velocity Resolution refers to the ability of the radar to distinguish targets and the required minimum velocity between them to do so.

APPENDIX B

Paper

The following article has been accepted for publication in EAI MobiQuitous 2021, the International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (November 8-11, 2021).

Radar-Based Gesture Recognition Towards Supporting Communication in Aphasia: The Bedroom Scenario*

Luís Santana, Ana Patrícia Rocha, Afonso Guimarães, Ilídio C. Oliveira, José Maria Fernandes, Samuel Silva, and António Teixeira

Institute of Electronics and Informatics Engineering of Aveiro, Department of Electronics, Telecommunications and Informatics, University of Aveiro, Aveiro, Portugal
{luis.santana, aprocha, afonso.guima, ico, jfernan, sss, ajst}@ua.pt

Abstract. Aphasia and other communication disorders affect a person's daily life, leading to isolation and lack of self-confidence, affecting independence, and hindering the ability to express themselves easily, including asking for help. Even though assistive technology for these disorders already exists, solutions rely mostly on a graphical output and touch, gaze, or brain-activated input modalities, which do not provide all the necessary features to cover all periods of the day (e.g., night-time). In the scope of the AAL APH-ALARM project, we aim at providing communication support to users with speech difficulties (mainly aphasics), when lying in bed. Towards this end, we propose a system based on gesture recognition using a radar deployed, for example, in a wall of the bedroom. A first prototype was implemented and used to evaluate gesture recognition, relying on radar data and transfer learning. The initial results are encouraging, indicating that using a radar can be a viable option to enhance the communication of people with speech difficulties, in the in-bed scenario.

Keywords: Smart environments · Communication · Gestures · FMCW radar · In-bed scenarios · Aphasia.

1 Introduction

People suffering from communication impairments have much more difficulty expressing their needs in ways that other people can understand. These difficulties can lead to problems socialising and limit the person's independence, namely in asking for help when needed.

Existing assistive technology for augmenting or replacing speech includes devices providing a graphical interface and relying on non-verbal interaction modalities, such as touch, gaze, or brain-activated, together with speech-generation [2].

* This work was supported by EU and national funds through the Portuguese Foundation for Science and Technology (FCT), in the context of the project AAL APH-ALARM (AAL/0006/2019) and funding to the research unit IEETA (UIDB/00127/2020).

These solutions require interacting with a given device (e.g., tablet), which may not be easily reached in some situations (e.g., lying in bed), rely on the use of cameras, which raises privacy concerns, and/or are too intrusive.

An alternative approach for assisting communication at a distance is the use of gestures. However, most contributions focus specifically on sign language [8]. Gesture recognition has also been explored in the context of human-computer interaction, relying on wearable sensors [5], vision-based sensors [7], or radars [1,3,9,4]. The latter have advantage of being the less intrusive and also preserving the user's privacy.

The ongoing project APH-ALARM aims at allowing people suffering from aphasia (e.g., after a stroke) to communicate more easily with other people anywhere and anytime. In the scope of this project, our main objective is to enhance communication for people with speech difficulties, in the in-bed scenario (i.e., user lying in bed).

Towards this goal, we propose a system based on a Frequency Modulated Continuous Wave (FMCW) radar for supporting communication through gestures, in the considered scenario. A first prototype, where gesture recognition is performed by a model obtained through transfer learning and radar data, was developed to explore the viability of the technology. To the best of our knowledge, gesture interaction for the in-bed scenario, where some patients may spend a large part of their time, has not yet deserved much attention.

2 Radar-Based Gesture Recognition System

We propose the architecture of a system that aids communication when the user is alone in a bedroom, lying in bed, and may need to communicate with other people (e.g., caregiver, family member) to ask for help, for instance. As a first step towards a novel communication support system for patients with aphasia, we present a first prototype for radar-based gesture recognition.

2.1 General Architecture

An overview of the system is depicted in Fig. 1. A radar captures data from the detected targets, in this case the human body. These data are sent to a processing unit, where they are pre-processed by removing outliers. Features are then extracted and used to recognize the gesture being carried out. A final decision is made and sent to a smartphone.

2.2 First Prototype

As a proof-of-concept, we implemented a first prototype that relies on the setup shown on the left side of Fig. 1, which includes a bed and a radar. The radar is elevated 0.55 m from the ground, and placed at 1 m from the bed, on the left side of the subject, parallel to the longest side of the bed. The radar's 2D coordinate system is shown in Fig. 1.

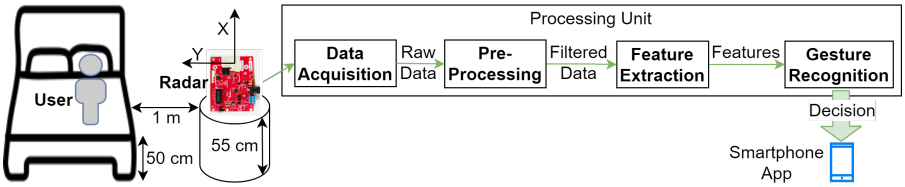


Fig. 1: Overview of the proposed system, including a possible setup for the bed and radar, as well as the pipeline for gesture recognition.

The radar is a Frequency Modulated Continuous-Wave (FMCW) radar from Texas Instruments, the AWR1642, with notable configurations entailing: frame rate (20 fps); resolution for range (4 cm) and radial velocity (0.22 m/s); maximum range (10.28 m) and radial velocity (3.47 m/s); for object detection, peak grouping in the Doppler (on) and range (off) directions, without clutter removal.

Data Acquisition: The data provided by the radar includes the X and Y coordinates, as well the Doppler index, for each detected moving target. In this first prototype, the data captured by the radar are saved to a computer.

Pre-Processing: The acquired data are processed using a sliding window of 5 s without overlap. For each window, pre-processing consists of removing outliers corresponding to unwanted reflections or noise. A detected target is considered as an outlier if its Euclidean distance to the radar is outside the interval [0.5, 3] m or its absolute Doppler index is outside the interval [1e−5, 10]. All data samples with X and Y coordinates outside the intervals [-1.5, 1.5] m and [0, 2.25] m, respectively, were also discarded.

Feature Extraction: From the filtered data, three different maps are created, one for each data type versus the elapsed time (X-Time, Y-Time, and Doppler-Time). The beginning and ending of the window where no movement is detected are discarded. An example of the X-Time and Doppler-Time maps for a repetition of the third gesture described below ("Back and Forth") is presented in Fig. 2, where the colour represents the number of detected targets (bright yellow corresponds to the maximum value for each map, while dark blue corresponds to no detected target). The matrix associated to each map is used to obtain a normalized greyscale image. The three images are then combined into a single image (X-Time above Y-Time above Doppler-Time).

Gesture Recognition: The images resulting from feature extraction are fed into a model that performs gesture recognition. This model is previously trained using the transfer learning method, relying on a pre-trained deep neural network model for image classification, and a given dataset. For this prototype, the focus was on the recognition of the following three arm gestures, all starting with the arm parallel to the body and resting on the bed: (a) **Wave** – Move the arm and hand from left to right and back, starting with the arm parallel to the body; (b) **Raise Arm** – Raise the arm until a 90°angle is formed with the body and then lower it back to initial position; (c) **Back and Forth** – "Come to me" motion, where the forearm is moved towards the arm making an angle

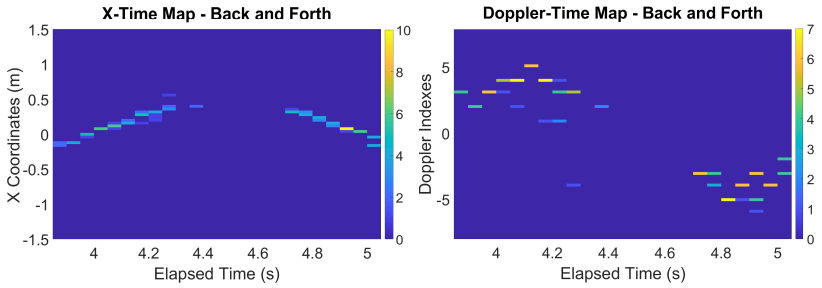


Fig. 2: Example of X-Time (left) and Doppler-Time (right) maps, for the "Back and Forth" gesture.

below 90°, and then returning to full extension. These gestures were selected aiming at simplicity and based on initial feedback from therapists and carers on the gestures' suitability for aphasic patients lying in bed. Moreover, they can be used for generating simple messages (e.g., "I need help") and "Yes/No" answer.

3 Evaluation

An initial evaluation of the prototype was performed to explore the possibility of recognising the defined gesture set using radar data together with transfer learning, in the context of the in-bed scenario.

Subject and Experimental Protocol: Radar data were captured from a 23-year-old, right-handed male subject. The used setup is the one included in Fig. 1, where the subject was lying in bed on their back. Each considered gesture was executed 50 times. Even though the subject is right-handed, all gestures were performed with the left arm, due to the radar being on the left side of the bed. For each repetition, data recording was initiated before the gesture execution and stopped automatically after 5 seconds.

Datasets: The obtained dataset includes 150 images (50 per gesture). Since deep learning requires a large dataset to obtain reasonable results, we expanded the dataset relying on offline data augmentation, to obtain a better performance and avoid overfitting. For each image in the original dataset, 5 or 10 new images were created by adding noise to that image (resulting in two augmented datasets). The type of noise added to the image is chosen randomly and can be a combination of the following types: Gaussian, salt and pepper, and Poisson. For all except Poisson, the amount of noise was limited to a proportion of image pixels to replace of 0.002 (chosen empirically).

Gesture Recognition Models: To obtain a model that recognises the considered gestures, we used the transfer learning method. Since our aim is to run gesture recognition in a processing unit with limited memory and computing capability, from the pre-trained models directly available in Keras [6], we explored three that achieved a top-5 accuracy equal or greater than 90% (ImageNet validation dataset) and have less than 10 million parameters: MobileNetV2, NASNetMobile, DenseNet121. For each pre-trained model, the top

layers were replaced by a single fully connected layer with 256 neurons (ReLU as the activation function) and an output layer with 3 neurons (softmax activation function). The used optimizer was ADAM (default parameters). Crossentropy was used as the loss function, and accuracy as the metric to be evaluated during training and validation.

Evaluation Method: Each model was evaluated using the 10-fold cross-validation approach, where 80% of the dataset is used for training, 10% for validation, and 10% for testing, in each iteration. Training is stopped when the validation loss has not decreased more than 0.1 for 5 epochs. The resulting model is evaluated on the test data of the corresponding iteration.

4 First Results

Results were obtained for the three pre-trained models listed above and for three different datasets: original (150 images); augmented 1 (750 images); augmented 2 (1500 images). The boxplots for the accuracy, F1 score, train time, and prediction time per image, considering all 10 folds, are shown in Fig. 3.

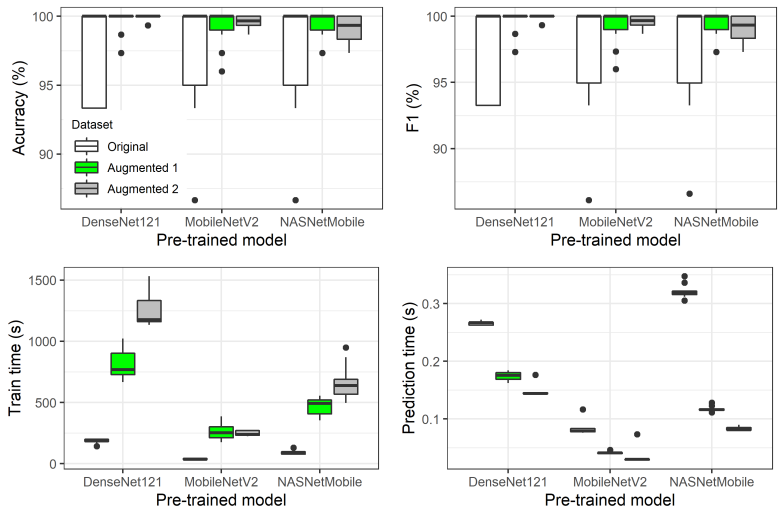


Fig. 3: Boxplots for the accuracy (left-top), F1 score (right-top), train time (left-bottom), and prediction time (right-bottom), for each model and dataset.

We can see that augmenting the data has an overall positive effect when it comes to the variability of accuracy and F1 score, for all three models. On the other hand, the train time increases when the size of the dataset increases, as expected, but the prediction time per image decreases. For both train and prediction times, the difference among datasets is lower for the MobileNetV2 model, which also has the lowest median train and prediction times: 35 to 252 s and 0.03 to 0.08 s, respectively, versus 84 to 639 s and 0.08 to 0.32 s for NASNetMobile, and 185 to 1177 s and 0.14 to 0.27 s for DenseNet121. This

was also expected, since MobileNetV2 is the smallest of the three pre-trained models (≈3.5 M parameters), followed by NASNetMobile (≈5.3 M parameters; DenseNet21 has ≈8.1 M). Despite its smaller size, MobileNetV2 still leads to a model with a median accuracy and F1 score similar to the other models (≥99% for all datasets). Although these results are quite good, it can be because only three gestures were considered and all used data came from the same subject.

5 Conclusion and Future Work

Our long-term research goal is the implementation of gesture-based communication support system for people with speech difficulties, such as aphasics. This system would provide its users with a more assisted and independent life, including at night-time. Our initial results on gesture recognition are in line with those reported in other similar contributions using radar (in scenarios different from the in-bed setting) [3,9,4]. They show the feasibility of recognising a simple set of gestures, in the specific in-bed scenario, based on a radar, which is not invasive or intrusive and can be placed in the environment.

Our study has some limitations, such as a small dataset limited to one subject and three gestures. However, we intend to obtain a larger dataset including more gestures and data from a greater number of subjects. This dataset will allow us to investigate if a model trained with data from a given subject can be used to recognise gestures performed by never seen subjects.

References

1. Ahmed, S., Kallu, K.D., Ahmed, S., Cho, S.H.: Hand gestures recognition using radar sensors for human-computer-interaction: A review. Remote Sensing **13**(3) (2021)
2. Elsahar, Y., Hu, S., Bouazza-Marouf, K., Kerr, D., Mansor, A.: Augmentative and Alternative Communication (AAC) advances: A review of configurations for individuals with a speech disability. Sensors **19**(8) (Apr 2019)
3. Hazra, S., Santra, A.: Robust gesture recognition using millimetric-wave radar system. IEEE Sensors Letters **2**(4), 1–4 (2018)
4. Ishak, K., Appenrodt, N., Dickmann, J., Waldschmidt, C.: Human gesture classification for autonomous driving applications using radars. In: IEEE MTT-S Int. Conference on Microwaves for Intelligent Mobility (ICMIM). pp. 1–4 (2020)
5. Jiang, S., Kang, P., Song, X., Lo, B., Shull, P.B.: Emerging wearable interfaces and algorithms for hand gesture recognition: A survey. IEEE Reviews in Biomedical Engineering pp. 1–1 (2021)
6. Keras: Keras applications, https://keras.io/api/applications/
7. Wang, T., Li, Y., Hu, J., Khan, A., Liu, L., Li, C., Hashmi, A., Ran, M.: A survey on vision-based hand gesture recognition. In: Basu, A., Berretti, S. (eds.) Smart Multimedia. pp. 219–231. Springer International Publishing, Cham (2018)
8. Yasen, M., Jusoh, S.: A systematic review on hand gesture recognition techniques, challenges and applications. PeerJ Comput. Sci. **5**, e218 (Sep 2019)
9. Yu, M., Kim, N., Jung, Y., Lee, S.: A frame detection method for real-time hand gesture recognition systems using CW-radar. Sensors **20**(8) (2020)

Bibliography

[1] A. Rahul and J. Winston, "Professionals' representation of communication disorders," *2020 IJRAR (InternationL Journal of Research and Analytical Reviews) December 2020, Volume 7, Issue 4*, Dec. 2020.

[2] National Institute on Deafness and Other Communication Disorders. (Acessed in: January 2021). "Aphasia statistics," [Online]. Available: https://www.nidcd.nih.gov/health/statistics/statistics-voice-speech-and-language.

[3] American Speech-Language-Hearing Association. (Acessed in: January 2021). "Communication disorders," [Online]. Available: https://www.asha.org/policy/RP1993-00208/.

[4] B. Bronken, M. Kirkevold, R. Martinsen, T. Wyller, and K. Kvigne, "Psychosocial well-being in persons with aphasia participating in a nursing intervention after stroke," *Nursing research and practice*, vol. 2012, p. 568 242, Jul. 2012. DOI: 10.1155/2012/568242.

[5] National Aphasia Association. (Acessed in: January 2021). "Aphasia fact sheet," [Online]. Available: https://www.aphasia.org/aphasia-resources/aphasia-factsheet/.

[6] Y. Elsahar, S. Hu, K. Bouazza-Marouf, and A. Mansor, "Augmentative and Alternative Communication (AAC) Advances: A Review of Configurations for Individuals with a Speech Disability," *Sensors 19(8):1911*, Apr. 2019. DOI: 10.3390/s19081911.

[7] A. Guimarães, A. P. Rocha, L. Santana, I. C. Oliveira, J. M. Fernandes, S. Silva, and A. Teixeira, "Enhanced communication support for aphasia using gesture recognition: The bedroom scenario," in *2021 IEEE International Smart Cities Conference (ISC2)*, 2021, pp. 1–4. DOI: 10.1109/ISC253183.2021.9562810.

[8] ——, "Wearable solutions towards gesture-based communication for aphasics while in bed," 2021.

[9] National Institute on Deafness and Other Communication Disorders. (Acessed in: January 2021). "What is aphasia?" [Online]. Available: https://www.nidcd.nih.gov/health/aphasia.

[10] M. B. Johansson, M. Carlsson, and K. Sonnander, "Communication difficulties and the use of communication strategies: From the perspective of individuals with aphasia," *International Journal of Language and Communication Disorders*, Mar. 2012. DOI: 10.1111/j.1460-6984.2011.00089.x.

[11] Tobii Dynavox. (Acessed in: January 2021). "I-Series | eye tracking-enabled SGD | world's nr°1 eye tracker - Tobii Dynavox," [Online]. Available: https://www.tobiidynavox.com/products/i-series/.

[12] Forbes AAC. (Acessed in: January 2021). "WinSlate 12D™ | Forbes AAC | Augmentative Communication," [Online]. Available: https://www.forbesaac.com/winslate-12d.

[13] ——, (Acessed in: January 2021). "WinSlate 12D™ with Enable Eyes™ | Forbes AAC," [Online]. Available: https://www.forbesaac.com/winslate-12-enable-eye%22.

[14] AssistiveWare. (Acessed in: January 2021). "Proloquo2Go - AAC app with symbols - AssistiveWare," [Online]. Available: https://www.assistiveware.com/products/proloquo2go.

[15] Therapy Box. (Acessed in: January 2021). "Predictable | Therapy Box," [Online]. Available: https://www.assistiveware.com/products/proloquo2go.

[16] BrainGate. (Acessed in: January 2021). "Assistive Communication | BrainGate," [Online]. Available: https://www.braingate.org/research-areas/assistive-communication/.

[17] C. Brandenburg, L. Worrall, A. Rodriguez, and D. Copland, "Mobile computing technology and aphasia: An integrated review of accessibility and potential uses," *Aphasiology*, vol. 27, pp. 444–461, Apr. 2013. DOI: 10.1080/02687038.2013.772293.

[18] J. Morris, J. Mueller, and M. Jones, "Toward mobile phone design for all: Meeting the needs of stroke survivors," *Topics in stroke rehabilitation*, vol. 17, pp. 353–61, Sep. 2010. DOI: 10.1310/tsr1705-353.

[19] M. Russo, V. Prodan, N. Meda, L. Carcavallo, A. Muracioli, L. Sabe, L. Bonamico, R. Allegri, and L. Olmos, "High-technology augmentative communication for adults with post-stroke aphasia: A systematic review," *Expert Review of Medical Devices*, vol. 14, Apr. 2017. DOI: 10.1080/17434440.2017.1324291.

[20] H. Liu and L. Wang, "Gesture recognition for human-robot collaboration: A review," *International Journal of Industrial Ergonomics*, vol. 68, pp. 355–367, 2018, ISSN: 0169-8141. DOI: https://doi.org/10.1016/j.ergon.2017.02.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169814117300690.

[21] O. Köpüklü, T. Ledwon, Y. Rong, N. Kose, and G. Rigoll, "Drivermhg: A multi-modal dataset for dynamic recognition of driver micro hand gestures and a real-time recognition framework," in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, 2020, pp. 77–84. DOI: 10.1109/FG47880.2020.00041.

[22] P. Molchanov, S. Gupta, K. Kim, and K. Pulli, "Multi-sensor system for driver's hand-gesture recognition," in *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, vol. 1, 2015, pp. 1–8. DOI: 10.1109/FG.2015.7163132.

[23] J. Lien, N. Gillian, M. E. Karagozler, P. Amihood, C. Schwesig, E. Olson, H. Raja, and I. Poupyrev, "Soli: Ubiquitous Gesture Sensing with Millimeter Wave Radar," *ACM Trans. Graph.*, vol. 35, no. 4, Jul. 2016, ISSN: 0730-0301. DOI: 10.1145/2897824.2925953. [Online]. Available: https://doi.org/10.1145/2897824.2925953.

[24] S. Hazra and A. Santra, "Robust Gesture Recognition Using Millimetric-Wave Radar System," *IEEE Sensors Letters*, vol. 2, no. 4, pp. 1–4, 2018. DOI: 10.1109/LSENS.2018.2882642.

[25] T. Schlömer, B. Poppinga, N. Henze, and S. Boll, "Gesture recognition with a wii controller," in *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction*, ser. TEI '08, Bonn, Germany, 2008, pp. 11–14, ISBN: 9781605580043. DOI: 10.1145/1347390.1347395. [Online]. Available: https://doi.org/10.1145/1347390.1347395.

[26] J. Park and S. H. Cho, "IR-UWB Radar Sensor for Human Gesture Recognition by Using Machine Learning," in *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2016, pp. 1246–1249. DOI: 10.1109/HPCC-SmartCity-DSS.2016.0176.

[27] J. Oh, T. Kim, and H. Hong, "Using Binary Decision Tree and Multiclass SVM for Human Gesture Recognition," in *2013 International Conference on Information Science and Applications (ICISA)*, 2013, pp. 1–4. DOI: 10.1109/ICISA.2013.6579388.

[28] J. Mlích, "Wiimote gesture recognition," in *Proceedings of the 15th Conference and Competition STUDENT EEICT*, Faculty of Electrical Engineering and Communication BUT, vol. 4, 2009, pp. 344–349.

[29] S. S. Rautaray and A. Agrawal, "Interaction with virtual game through hand gesture recognition," in *2011 International Conference on Multimedia, Signal Processing and Communication Technologies*, 2011, pp. 244–247. DOI: 10.1109/MSPCT.2011.6150485.

[30] P.-H. Chou, Y.-L. Hsu, W.-L. Lee, Y.-C. Kuo, C.-C. Chang, Y.-S. Cheng, H.-C. Chang, S.-L. Lin, S.-C. Yang, and H.-H. Lee, "Development of a smart home system based on multi-sensor data fusion technology," in *2017 International Conference on Applied System Innovation (ICASI)*, 2017, pp. 690–693. DOI: 10.1109/ICASI.2017.7988519.

[31] S. Yang, S. Lee, and Y. Byun, "Gesture recognition for home automation using transfer learning," in *2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, vol. 3, 2018, pp. 136–138. DOI: 10.1109/ICIIBMS.2018.8549921.

[32] S. Machado and S. Mancheno, "Automotive FMCW Radar Development and Verification Methods," 2018.

[33] G. Galati and P. van Genderen, "History of radar: The need for further analysis and disclosure," in *2014 11th European Radar Conference*, 2014, pp. 25–28. DOI: 10.1109/EuRAD.2014.6991198.

[34] M. Guarnieri, "The Early History of Radar," *Industrial Electronics Magazine, IEEE*, vol. 4, pp. 36–42, Oct. 2010. DOI: 10.1109/MIE.2010.937936.

[35] Q. Wan, Y. Li, C. Li, and R. Pal, "Gesture recognition for smart home applications using portable radar sensors," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014, pp. 6414–6417. DOI: 10.1109/EMBC.2014.6945096.

[36] C. Li, V. M. Lubecke, O. Boric-Lubecke, and J. Lin, "A Review on Recent Advances in Doppler Radar Sensors for Noncontact Healthcare Monitoring," *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, no. 5, pp. 2046–2060, 2013. DOI: 10.1109/TMTT.2013.2256924.

[37] Y. Xiao, J. Lin, O. Boric-Lubecke, and V. Lubecke, "A ka-band low power doppler radar system for remote detection of cardiopulmonary motion," in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, 2005, pp. 7151–7154. DOI: 10.1109/IEMBS.2005.1616156.

[38] A. Droitcour, V. Lubecke, J. Lin, and O. Boric-Lubecke, "A microwave radio for Doppler radar sensing of vital signs," in *2001 IEEE MTT-S International Microwave Sympsoium Digest*, vol. 1, 2001, 175–178 vol.1. DOI: 10.1109/MWSYM.2001.966866.

[39] S. Lee, Y.-J. Yoon, J.-E. Lee, and S.-C. Kim, "Human–vehicle classification using feature-based SVM in 77-GHz automotive FMCW radar," *IET Radar, Sonar & Navigation*, vol. 11, no. 10, pp. 1589–1596, 2017. DOI: https://doi.org/10.1049/iet-rsn.2017.0126. eprint: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-rsn.2017.0126. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-rsn.2017.0126.

[40] H. Zhou, P. Cao, and S. Chen, "A novel waveform design for multi-target detection in automotive FMCW radar," in *2016 IEEE Radar Conference (RadarConf)*, 2016, pp. 1–5. DOI: 10.1109/RADAR.2016.7485315.

[41] C. Li, C. Gu, R. Li, and S. B. Jiang, "Radar motion sensing for accurate tumor tracking in radiation therapy," in *WAMICON (Wireless and Microwave Technology Conference) 2011*, 2011, pp. 1–6. DOI: 10.1109/WAMICON.2011.5872871.

[42] E. Hyun and J.-H. Lee, "A meothod for multi-target range and velocity detection in automotive FMCW radar," in *2009 12th International IEEE Conference on Intelligent Transportation Systems*, 2009, pp. 1–5. DOI: 10.1109/ITSC.2009.5309873.

[43] B. Dekker, S. Jacobs, A. Kossen, M. Kruithof, A. Huizing, and M. Geurts, "Gesture recognition with a low power fmcw radar and a deep convolutional neural network," in *2017 European Radar Conference (EURAD)*, 2017, pp. 163–166. DOI: 10.23919/EURAD.2017.8249172.

[44] S. Ahmed, K. D. Kallu, S. Ahmed, and S. H. Cho, "Hand gestures recognition using radar sensors for human-computer-interaction: A review," *Remote Sensing*, vol. 13, no. 3, 2021, ISSN: 2072-4292. DOI: 10.3390/rs13030527. [Online]. Available: https://www.mdpi.com/2072-4292/13/3/527.

[45] BuiltIn. (Acessed in: October 2021). "A Step-by-Step Explanation of Principal Component Analysis (PCA)," [Online]. Available: https://builtin.com/data-science/step-step-explanation-principal-component-analysis.

[46] Keras, *Keras applications*, Acessed in: January 2021. [Online]. Available: https://keras.io/api/applications/.

[47] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520. DOI: 10.1109/CVPR.2018.00474.

[48] T. I. Sandeep Rao. (Acessed in: June 2021). "Introduction to MMWave Sensing: FMCW Radars," [Online]. Available: https://training.ti.com/sites/default/files/docs/mmwaveSensing-FMCW-offlineviewing_0.pdf.

[49] S. R. Cesar Iovescu. (Acessed in: June 2021). “The fundamentals of millimeter wave sensors,” [Online]. Available: https://www.ti.com/lit/wp/spyy005a/spyy005a.pdf?ts=1601720176627&ref_url=https%253A%252F%252Fwww.ti.com%252Fsensors%252Fmmwave-radar%252Fwhat-is-mmwave.html.