Target Motion Reconstruction using FMCW radar for Hand Gestures

Prepared by:
Ethan Meknassi

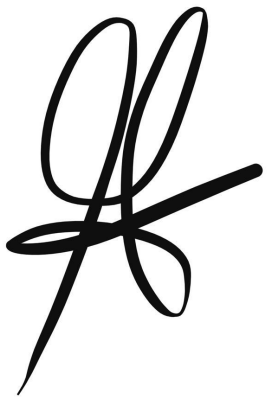
Prepared for:
Dr. Stephen Paine
Department of Electrical Engineering
University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town
in partial fulfilment of the academic requirements for a
BSc in Electrical and Computer Engineering

October 29, 2023

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.



October 29, 2023

Ethan Meknassi — Date

Word Count: 21844 words

Acknowledgements

I would like to express my gratitude to Dr Stephen Paine, my supervisor, for consistently offering support and guidance throughout the project. Thank you for pushing me to my limits during this journey.

I am also thankful to the entire RRSG group for their support during this semester and the countless hours spent together in the lab. I want to extend special thanks to Nicholas Bowden and Mark Nyago for their invaluable assistance and guidance at every step of the project. I wish all of you great success in your future endeavours. Thank you to Grant for providing guidance in some parts of the report.

My heartfelt appreciation goes out to my friends who accompanied me on this engineering journey, namely Zac, Jason, Jahar, Jonathan and Ryan. I am grateful for the good times and emotional support you provided, and your advice was always invaluable in navigating the challenges of this degree. I wish each of you immense success in your future.

I want to give a special mention to Ethan, who's been like a brother to me and shared this entire engineering journey with me. It would not have been the same without your presence, and I am thankful for everything you've done.

Finally, I extend my deepest gratitude to my parents, Penelope and Sebastien, for their immense love and support over the past four years. Thank you for believing in me. And to my girlfriend, Bianca, I am grateful for all the love and encouragement you've given me throughout this journey.

Abstract

Hand-gesture recognition (HGR) technology enables devices to identify and interpret human hand movements. This project introduces a HGR system using Millimeter Wave (mmWave) Frequency Modulated Continuous Wave (FMCW) radar technology for the recognition of distinct hand gestures. The project objectives encompass the creation of a hand gesture dataset, the acquisition of raw radar data, the development of a data processing pipeline, and the implementation of a machine learning model for gesture recognition. This report comprehensively documents the research process, including an extensive literature review, theoretical background, design process, and the obtained results. The hand gesture dataset consists of five distinct hand gestures, with a total of 750 samples collected from the author. The processing pipeline takes raw radar data as input and employs Range-Doppler Map (RDM), clutter removal techniques, and noise reduction methods to enhance the quality of the resulting radar maps. The output of the processing pipeline includes two types of radar maps: Range-Time Map (RTM) and Doppler-Time Map (DTM). For the recognition of hand gestures from raw radar data, a Convolutional Neural Networks (CNN) classification algorithm was selected. The two output maps are combined into a single image, which serves as input for the CNN. After training the CNN model, the proposed system was tested with unseen data. All project objectives were successfully met, resulting in a classification accuracy of 100% for all five types of hand gestures. This performance aligns with previous findings in the current state-of-the-art implementations.

Contents

List of Figures

Abbreviations

ATM Angle-Time Map

CFAR Constant False Alarm Rate

CNN Convolutional Neural Networks

CSI Channel State Information

CTC Connectionist Temporal Classification

CW Continuous Wave

DAM Doppler-angle Map

DTM Doppler-Time Map

DTW Dynamic Time Warping

EM Electromagnetic

FFT Fast Fourier Transform

FMCW Frequency Modulated Continuous Wave

GreBsmo Greedy bilateral smoothing

HCI Human-Computer Interaction

HDF5 Hierarchical Data Format version 5

HGR Hand-gesture recognition

HMM Hidden Markov Models

I3D Inflated 3D ConvNet

IMU Internal measurement units

IQ In-Phase Quadrature

ISM Industrial, Scientific, and Medical

KNN K-Nearest Neighbor

LSTM Long Short-Term Memory

LTRACN Long-Term Recurrent All-Convolutional Network

mmWave Millimeter Wave

NUC Next Unit of Computing

PRDM Processed Range-Doppler Map

RAM Range-Angle Map

RDM Range-Doppler Map

ReLu (Rectified Linear Unit)

REM Range-Elevation Map

RF Radio Frequency

RGB Red Green Blue

RNN Residual Neural Network

RTM Range-Time Map

S3D Spatiotemporal 3D

SLFG stochastic linear formal grammar

SNN Spiking Neural Networks

SNR Signal to Noise Ratio

STFT Short Time Fourier Transform

SVM Support Vector Machine

UWB Ultra-wideband

Wi-Fi Wireless Fidelity

YOLO You Only Look Once

Chapter 1

Introduction

1.1 Background of the project

Over the last decade, the practice of Hand-gesture recognition (HGR) has grown significantly due to the continuous development of Human-Computer Interaction (HCI) field. Recently, with the advancements of machine learning techniques, it has become even more popular. HGR is a technology which enables the identification and interpretation of human hand gestures. HGR enables humans to communicate with devices intuitively, offering a wide range of valuable applications, including Sign Language translation and remote device control, among others. Within this context, the use of Millimeter Wave (mmWave) Frequency Modulated Continuous Wave (FMCW) technology has gained significant attention due to its numerous advantages. Radar technology facilitates precise hand gesture detection, even in challenging environmental conditions.

1.2 Objectives

The primary objective of this study is to design and implement a HGR system utilising mmWave FMCW radar technology for the recognition of a predefined set of distinct hand gestures. This objective can be deconstructed into specific goals:

- Formulate a set of distinct hand gestures and gather raw radar data to establish a dataset.
- Designing and implementing a radar processing pipeline to transform raw radar data into a format that can be inputted into a machine learning algorithm.
- Create a machine learning algorithm for recognising the designed hand gestures as a proof-of-concept.

1.3 Application Scope

In order to achieve the objectives mentioned above, this project entails the investigation, design, and implementation of a HGR system utilising mmWave FMCW technology. The scope of the project encompasses the following key elements:

- A comprehensive review of prior research methodologies and findings.
- A discussion of the theory used throughout the project.

- In-depth analysis regarding major design choices, including a selection process for each choice.
- The creation of a dataset for hand gestures.
- The acquisition of raw radar data corresponding to the designed hand gestures.
- The design and implementation of a data processing pipeline.
- The development and implementation of a machine learning algorithm, serving as a proof-of-concept for the entire system.
- A series of experiments and subsequent analysis aimed at evaluating the performance of the proposed system as a whole.

1.4 Limitations

This project is subject to several significant constraints. To begin with, as with any engineering problem, time has played a crucial role in the design of the proposed system. The process of gathering raw radar data for hand gestures can be quite time-consuming, particularly when dealing with extensive datasets. Given the project's constrained timeline of approximately thirteen weeks, it is not feasible to perform multiple rounds of data collection. Consequently, the system is restricted to implementing only five specific hand gestures and does not encompass all the common hand gestures.

Furthermore, due to circumstances regarding ethical clearance, the project's data collection is limited to a single participant. This limitation may affect the diversity of the hand gesture dataset.

The project's budget is limited to ZAR2000, which has played a vital role in the hardware selection process. The choice of hardware components has been made based on their availability and feasibility, and the radar employed was restricted to the equipment accessible in the radar laboratory.

Additionally, due to both time and budgetary constraints, the proposed system is developed as a prototype in which each subsystem must be operated independently and sequentially. Consequently, the system is limited to non-real-time HGR.

Lastly, the acquisition of raw hand gesture data took place within the radar laboratory. Consequently, the system's performance may be susceptible to additional challenges arising from possible undesired signals within the environment.

1.5 Report Outline

Following this introductory chapter, the report proceeds to chapter 2, which offers a comprehensive review of past research methodologies and findings in the field of HGR using mmWave FMCW radar technology. The report then follows with chapter 3 by providing the theoretical background that was utilised during the design process. Chapter 4, describes the system design, encompassing the procedural steps and various choices made throughout the project. Subsequently, chapter 5, presents the results derived from the distinct project stages, accompanied by an analysis and evaluation of these outcomes. The report concludes with Chapter 6, offering a conclusions of the entire project, along with future recommendations for the project.

Chapter 2

Literature Review

The following chapter in this report delves into the background of HGR using mmWave FMCW radar technology. It explores the accomplishments of previous research in this field and discusses the various existing methods.

This literature review initiates with a discussion of the history of HGR, providing historical context to the project. It proceeds to analyse the various stages involved in the system's implementation, delving into the choices, methods, and outcomes found in the current state-of-the-art. Furthermore, it explores contemporary applications of HGR. The literature review concludes by addressing potential limitations identified in prior research.

2.1 History of Hand Gesture Recognition

This section offers a review of the significant historical developments in HGR. It commences with an examination of the first HGR development and subsequently delves into the evolution of this technology over the past decade. The various approaches in HGR are discussed, including wearable devices and motion-based sensors, vision-based techniques, Wi-Fi signal analysis, and radar-based approaches. Additionally, the advantages and limitations of each approach is detailed.

The earliest implementation of this technology dates back to 1987 with the development of a hand to machine interface by Thomas G. Zimmerman et al., called the DataGlove [1]. This device tracks hand movements, finger bending, and hand position, using five to fifteen analogue flex sensors, magnetic flux sensors, and ultrasonics. The sensors are mounted directly onto the glove and wired to the host computer, allowing for precise measurement of the spatial position and orientation of the hand.

This technology, based on motion sensors and Internal measurement units (IMU), has continued to evolve since then. It involves using accelerometers or gyroscopes on a wearable device and has been employed in various challenging applications, such as sign language translation and air-writing [2–5]. However, these devices are not available for individuals with impairments or injuries. Therefore, a prototype was developed by Singh et al. in 2015 that used textile capacitive arrays to overcome these limitations [6]. Despite these advancements, the main drawbacks of motion-based sensors are that wearable HGR technology can be expensive and that the user must wear the device, which limits its flexibility and functionality.

As a result, contactless techniques have gained more attention in recent years. One common technique

is vision-based sensors, which employ image processing techniques to recognise hand gestures. This approach commonly utilises Red Green Blue (RGB) images and depth descriptors [7–9], while others use depth camera systems like Intel's RealSense [10] or Kinect [11]. Although these sensors have excellent performance for HGR due to their high resolution, they have significant limitations. They are unable to detect through objects and are sensitive to ambient lighting conditions. Furthermore, this type of sensor does not preserve privacy and can sometimes have high power consumption.

Another approach involves HGR based on Wireless Fidelity (Wi-Fi) signals, which utilises Electromagnetic (EM) signals for communication [12–15]. This method analyses the variations in waveform patterns within the Channel State Information (CSI) of Wi-Fi signals, induced by hand gestures, to identify them as distinctive HGR features [14]. However, this approach has limitations due to its susceptibility to interference. The waveforms in Wi-Fi signals are originally designed for communication purposes, which means that this method may not deliver accurate results in unsuitable environmental conditions [15].

Finally there exists radar-based methods, which has gained significant attention in industry and academia due to its numerous advantages. This method can detect and classify hand gestures even in poor ambient lighting conditions and stay accurate even in the presence of obstructions. Additionally, this approach does not require any contact with the user, making it more convenient. Moreover, radar-based methods can perform HGR even through objects, making the applications more versatile. In 2020, Xia et al. [16] conducted research highlighting different approaches within radar-based methods, including Doppler-radar-based, Ultra-wideband (UWB) radar-based, and FMCW radar-based. The Doppler-radar-based method detects micro-Doppler signatures caused by EM signals reflected from the moving hand [17, 18]. However, it has limitations due to its low-range resolution and poor ability to negate interferences. The UWB radar-based method utilises time differences between the hand signals and the transmitted signal to detect hand gestures [19, 20]. Research by Skaria et al. in 2020 found that UWB radar-based methods for HGR have advantages such as minimal power usage, precise range resolution, and the capability to detect targets in close proximity. Despite this, they can be disadvantageous due to the high-performance requirements for signal processing and modulation, leading to the need for high-cost processors. In comparison, FMCW radar can detect hand gestures in a simpler way and at a lower cost. The applications of FMCW radar-based methods have higher range, time, and velocity resolution, making this approach the most effective when it comes to HGR. This can be highlighted by the first and most popular implementation of HGR, which utilised mmWave FMCW radar in 2016, the Google Soli project [21].

2.2 Implementation of Hand Gesture Recognition using mmWave FMCW Radar

In the realm of HGR systems, various radar technologies, types of gestures, processing techniques and classification algorihtm have been explored in research. However, there exists a consistent and well-defined process for HGR systems. This process comprises three distinct stages: the initial extraction of raw hand gesture data from the mmWave FMCW radar system, subsequent signal processing of the raw radar data, and the final step of feature extraction to prepare the data for a gesture classification algorithm. This systematic approach can be visualised through the following diagram:

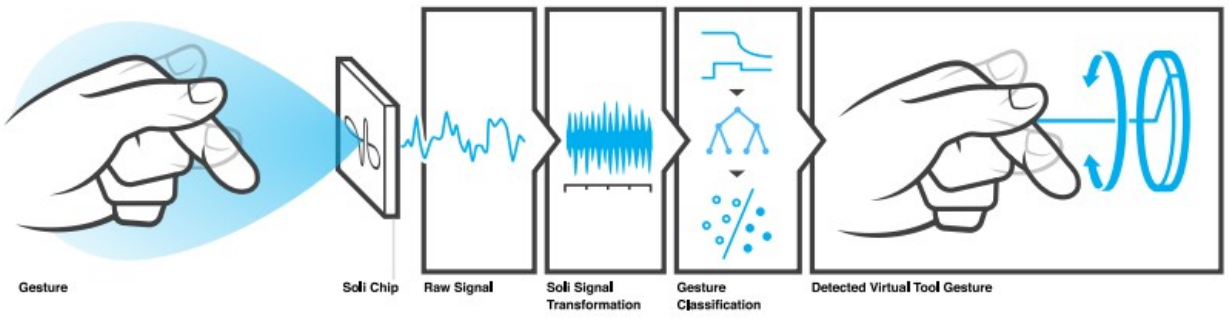


Figure 2.1: End-to-end HGR process using mmWave FMCW radar according to the Soli Project [21]

This section provides a review of the relevant literature concerning these three stages of HGR using mmWave FMCW radar. Beginning with an exploration of the raw radar signal stage, including discussions on the types of hand gestures studied, the available datasets, and the specific mmWave FMCW radar utilised. Subsequently, the raw radar data processing stage is then discussed, examining various processing techniques employed in the current state-of-the-art. Finally, past methodologies employed for classification algorithms in prior research are discussed. It is essential to recognise that each of these stages plays a crucial role in the average accuracy of the HGR system. Hence, each research study should be evaluated in its entirety. This is reflected in Subsection 2.3 which contains all the relevant research in the current state-of-the-art.

2.2.1 Hand Gesture Dataset and FMCW Radar Systems

Types of Gestures

In the domain of HGR systems, the specific type of gesture to be recognised holds a pivotal role, significantly influencing recognition accuracy. The choice of gestures for classification typically varies depending on the study's context and intended application. The inherent characteristics of each gesture play a substantial role in determining the performance requirements of the system [22]. Generally, larger gestures provide richer movement data, which can be translated into features that the system can more easily detect. Conversely, smaller movements may introduce increased difficulty in distinguishing features or targets within the raw radar data. Additionally, the distance at which hand gestures are performed relative to the radar device must be considered. Previous HGR research has identified an optimal range, typically spanning from 10 centimetres to 1 meter, for the distance between the gesturing individual and the radar system.

Historical research in HGR has categorised hand gestures into distinct types. Magrofuoco et al. [23] described four categories of hand gestures: microgestures, motion gestures, combined gestures, and 'other' gestures. Microgestures encompass subtle finger movements and articulation variations. Motion gestures involve changes in motion, often manifested as a change in the spatial position of the palm. Combined gestures merge both microgestures and motion gestures to create new gesture forms. Hand gestures that do not fall into these categories are typically classified as 'other' gestures.

Microgestures, which primarily rely on finger movements, can be more challenging to identify due to their limited surface area and, consequently, their restricted movement data. As a result, research and applications of HGR based on microgestures often demand higher requirements from the processing of the radar data and the classification algorithm. There exists several different microgestures which have been used in the past. Juaung et al. [24] conducted an HGR implementation centered on microgestures, requiring extensive pre-processing that involved a combination of processing techniques, a feature extraction phase, and classification algorithms in order to achieve a 98% accuracy rate in the classification model. This research encompassed various microgestures, such as clicking a virtual button, rotating a virtual disk, manipulating a virtual slider, and performing vertical and horizontal swipes. Additionally, a study by Li et al. [25] explored microgestures like 'finger tapping,' virtual slider manipulation, and finger waving, mimicking a slider. Different microgestures were also examined when Ali et al. [26] implemented an HGR system covering six intuitive hand gestures, including radial and tangential single-finger circles, single and double pinches, as well as horizontal swipes.

Motion-based hand gestures have gained popularity due to their relative ease of implementation compared to microgestures. Zhang et al. [27] introduced 'Lantern,' a Dynamic HGR system employing mmWave FMCW radar, which successfully detected and recognised eight distinct motion gestures, including right-to-left and left-to-right hand slides, pulling, pushing, knocking, vertical and horizontal hand movements, waving, and patting. Subsequently, Choi et al. [28] developed an HGR system featuring an array of hand gestures, incorporating actions such as clockwise and counterclockwise rotations, pushing, double pushing, drawing X and reverse-X, holding, and double-handed pushing.

In 2020, Yu et al. [29] implemented an HGR system aimed at enhancing HCI by incorporating hand gestures like grabbing, tilting the hand left and right, falling, making an arc, clapping, and lifting. Rodrigues and Li [30] explored large motion-based hand gestures, proposing four distinct hand movements: arm extension and flexion, hand sweeping toward the radar, horizontal hand rotation, and hand-clicking. This research was designed to investigate the feasibility of mmWave FMCW radar for HGR in scenarios involving multiple closely spaced targets. Furthermore, the study examined the random body motions of targets situated at the same range as the hand gesture, and affirmed the effectiveness of mmWave FMCW radar for HGR even under these challenging conditions.

Datasets for mmWave FMCW radar-based Hand Gesture Recognition

The adoption of mmWave FMCW radar for HGR has only gained popularity since 2016. As a result, most studies in the current state-of-the-art have relied on self-constructed datasets for the experiments. Previous research has highlighted that datasets encompassing a greater number of participants and a diverse array of hand gestures tend to have better generalisation. While it is true that larger datasets often necessitate more extensive and complex training procedures, they also tend to demonstrate

increased robustness. Typically, self-constructed datasets used in past studies contain a total sample data size ranging from 1000 to 5000 samples, although the specific numbers may vary across individual research endeavours.

Nevertheless, there are a few open-source resources available for radar-based hand gesture datasets. One notable example is the freely accessible Google Soli dataset, which features 2D-range-Doppler data presented as 32x32 images. This dataset comprises eleven distinct hand gestures, recorded from ten different participants, with each gesture being repeated 25 times, resulting in a total dataset size of 2750 samples [31]. Additionally, there exists some mmWave FMCW based datasets on online machine learning community websites such as Kaggle [32].

mmWave FMCW Radar

Over the past decade, various types of mmWave FMCW radars have been extensively explored and employed in the field of HGR. These radars have differences in their frequency band coverage, including 24GHz, 60GHz, and 77GHz, as well as variations in available bandwidth and the number of antennas.

The 24GHz and 60GHz frequency bands are commonly referred to as the Industrial, Scientific, and Medical (ISM) band. Within the 24GHz ISM band, radar systems operate within the frequency range of 24.0 GHz to 24.25 GHz, resulting in a limited bandwidth of 250MHz. Prominent radar chips operating in this frequency band include the BGT24MTR12 chip [27]. On the other hand, the 77GHz frequency band offers a more substantial available bandwidth of 4GHz. Texas Instruments, a renowned producer of radar-related hardware, has published a paper discussing the transition from 24GHz to 77GHz radar technology [33]. The company manufactures short-range radar chips such as the I/AWR1642BOOST, I/AWR1843BOOST, and I/AWR1443BOOST, all operating within the 77GHz-81GHz frequency band.

The performance of radar systems is typically evaluated based on parameters such as maximum range, maximum velocity, range resolution, and velocity resolution. As per radar theory, which will be further elaborated in Chapter 3, radar systems with greater available bandwidth offer better range resolutions, significantly enhancing their capability to distinguish between features or targets. Consequently, the 77-81GHz band has become the preferred choice for short-range radar in recent state-of-the-art research due to its 4GHz bandwidth, which yields superior range resolution compared to the 250MHz bandwidth of ISM band radars [33]. However, this does not undermine the relevance of radars operating in the 60GHz ISM band. Radar chips like the I/AWR6843, operating within the 60GHz to 64GHz frequency range, also provide a 4GHz bandwidth and offer similar benefits as their 77GHz counterparts.

The available bandwidth is not the sole factor influencing radar performance; the number of antennas can also have a significant impact on various aspects of radar functionality. Qu et al. [34] conducted a study in 2022 that emphasised the influence of using different numbers of receive antennas in HGR. Their research explored various configurations, by comparing the effects of using 1, 2 or 4 receive antennas and demonstrating that employing more receive antennas to collect hand gesture data results to higher recognition accuracy in classification models.

2.2.2 Raw Radar Data Processing

The raw radar data collected from hand gestures using the mmWave FMCW radar then undergoes subsequent processing. This stage of the systems enables the generation of maps from which various features related to different hand gestures can be extracted. These maps, which are often two-dimensional, convey information related to aspects such as distance, angle, and velocity about the target. Various types of maps can be derived from the raw radar data processing, including:

1. Range-Doppler Map (RDM) [35–37]
2. Range-Time Map (RTM) [27, 38, 39]
3. Doppler-Time Map (DTM) [40–42]
4. Angle-Time Map (ATM) [43–45]
5. Range-Angle Map (RAM) [29, 46, 47]
6. Range-Elevation Map (REM) [46]

Previous research have employed diverse maps for the processing of the raw radar data, each with their own set of advantages and disadvantages.

2.2.3 Feature Extraction and Classification Algorithms

Once the maps have been generated, their information becomes the input data for the classification algorithm. Past research have explored the use of a single map as the sole input for classification algorithms [35–38, 40, 41]. However, achieving high recognition accuracy with only one map as the input can be challenging. Machine learning classification models for HGR have shown that increasing the number of features can enhance accuracy, as observed in previous studies comparing single-feature versus multi-feature approaches [29, 45]. Despite some studies relying solely on a single feature, many have proposed methods to combine multiple maps, thereby incorporating a broader range of gesture characteristics (features) for the classification algorithm.

Research employing multiple features has typically followed two approaches: the use of multi-channel classification models or the combination of feature maps. Multi-channel classification models involve employing different types of information or images as inputs, with these maps often concatenated or merged at some point within the model's architecture [42–46, 48, 49]. Alternatively, some research have chosen to blend the maps to construct a new multi-dimensional feature structure known as a feature cube [50–53]. This structure combines and represents features across multiple dimensions, with the dimensions varying in size depending on the number of maps to be utilised.

Once the desired maps and features have been extracted for the model, the next crucial step involves selecting an appropriate classification algorithm. Given the wide array of classification algorithms available, previous HGR research has explored diverse approaches.

Deep learning-based classification algorithms have gained popularity in the current state-of-the-art for HGR. Neural network models, including Convolutional Neural Networks (CNN)s and Long Short-Term Memory (LSTM) networks, have emerged as prominent choices. Some research has even extended

standard CNNs to 3D-CNNs by incorporating an additional time dimension, often resulting in improved performance, despite the increased computational demands [44,51]. Other architectures like 2D and 3D Res-Net, Inflated 3D ConvNet (I3D), and Spatiotemporal 3D (S3D) models have also found application in HGR and frequently yield high performance due to their optimised neural network designs. However, it is worth noting that these models tend to be computationally more intensive compared to classic deep learning neural network models like CNNs.

Recent research have investigated the implementation of Spiking Neural Networks (SNN)s for hand gesture classification. SNNs utilise spikes to transmit information within the model. Additionally, the performance of simpler models such as K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) have also been investigated in the past. These algorithms, while relatively straightforward to implement, have often exhibited lower performance and less robustness compared to neural networks [53].

Hidden Markov Models (HMM) and Dynamic Time Warping (DTW) algorithm have also been applied to HGR. Lastly, some studies have explored hybrid approaches, such as combinations of the VGG19 model and an XGBoost Classifier, to further investigate the potential of different algorithms for HGR [49].

2.3 Related Works

Each research study in the field of mmWave FMCW radar-based HGR has variations in types of gestures, datasets, radar equipment, processing methods, feature extraction methods, and classification algorithms. Consequently, comparing the advantages and disadvantages of these different aspects is challenging. However, some studies have made comparisons while keeping certain criteria constant. This allows for the identification common patterns across various research efforts. This section presents an overview of the research conducted in mmWave FMCW radar for HGR, summarising the outcomes of various studies that have employed different methods. A table is provided at the end of the section for a clear and concise summary of the current literature findings.

2.3.1 Single Map Input

Beginning with the Google Soli project by Lien et al. [31] in 2016, this was the first mmWave FMCW radar system designed for HGR. It utilised a radar chip operating in the 60GHz frequency band, with a 7GHz available bandwidth and two transmit and four receive antennas. A single input method, made up of RDMs, was used for the classification algorithm. This project laid the foundation for subsequent HGR research using mmWave FMCW radar which started by following this single input method.

For example, Hazra and Santra [35] developed a robust HGR system using an mmWave FMCW radar, specifically the BGT60TR24 model, operating within the 60GHz frequency band with a 4GHz bandwidth. This radar had one transmit antenna and four receive antennas. The approach used involved employing RDMs as the sole feature for the classification algorithm. The investigation used a Long-Term Recurrent All-Convolutional Network (LTRACN), a fusion of CNN and LSTM, to recognise five distinct hand gestures. The training dataset consisted of 1500 samples collected from 10 different participants, with an additional test dataset of 600 sequences gathered from five other participants. The experiment yielded an average classification accuracy of 94.34%. The study also includes a comparison

between the proposed system and a CNN + LSTM end-to-end model, which achieved an accuracy of 84%, demonstrating the performance of the proposed approach.

In 2018, Suh et al. [54] developed a HGR system using a 24GHz mmWave FMCW radar system with an available bandwidth of 250MHz. This system employed just one transmit antenna and four receive antennas. It was specifically designed to recognise seven types of hand gestures performed by two participants, within a range of 0.2m to 0.4m and within ±30° of the radar's transmit antenna. The data collected from this radar system was processed into RDMs, which were used to generate Processed Range-Doppler Map (PRDM)s. The PRDMs involved 'averaging values in each range bin along the Doppler axis and averaging values in each Doppler bin along the range axis'. The PRDMs were then used to obtain the features for a classification algorithm, which achieved an average accuracy of over 91% in classifying the defined hand gestures. The training set of the model consisted of 840 sample data samples, while the test set contained 1960 sample data samples.

Moving to 2019, Choi et al. [28] also employed the same 60GHz radar chip Google Soli used and relied solely on RDMs as input for the HGR system. The dataset included ten hand gestures performed by ten participants for a total of 4000 hand gestures samples. Utilising a LSTM encoder, the system achieved an average classification accuracy of 99.1%. Additionally, the model exhibited an accuracy of 98.48% when tested on a new participant. The LSTM-based approach was compared to two other classification models: a Residual Neural Network (RNN) encoder and a 2D-CNN. The 2D-CNN and RNN achieved average accuracies of 95.39% and 92.9%, respectively. However, when tested on a new participant, these classification algorithms achieved lower average accuracies, specifically 74.70% for 2D-CNN and 85.93% for RNN.

In 2020, Zhang et al. [37] explored the performances of 3D-CNN and CNN-LSTM. The AWR1642BOOST radar was used to collect data for six different hand gestures performed by four participants, with 50 repetitions per hand gesture, totalling 1200 samples. RDMs were exclusively employed as the sole input for machine learning algorithms. The 3D-CNN model achieved an average accuracy of 95% and showed smoother training progression. However, after more training iterations, the CNN-LSTM model reached an average accuracy of 97%.

Similarly, Zhang et al. [27] demonstrated that high accuracy in HGR could be achieved using the BGT24MTR12 chip operating at the 24GHz frequency band if a bandwidth of 4GHz was used. The proposed system, named 'Lantern,' addressed common challenges in HGR, such as robustness in practical environmental conditions, variations in how individuals perform hand gestures, and real-time processing requirements. The BGT24MTR12 model used in this experiment included one transmit antenna and two receive antennas. This study featured eight different hand gestures, with a total dataset of 3200 samples collected from five different participants. The raw radar data was processed into RTMs, and a Connectionist Temporal Classification (CTC) algorithm was employed for HGR, with a focus on the temporal phase of the gesture. To further improve accuracy, a combination of a 3D-CNN and an LSTM encoder was used to analyse the hand gestures. The 'Lantern' system achieved an average accuracy of 96% in the classification of these hand gestures.

In 2023, Singam et al. [38] utilised Texas Instrument's IWR6843 mmWave radar, which operates within the 60GHz band and provides a 4GHz bandwidth, for HGR. This radar chip is equipped with three

transmit antennas and four receive antennas. Singam et al. constructed a dataset consisting of 500 samples for two distinct hand gestures targeted for recognition. The micro-Doppler signatures from the radar data were extracted and transformed into RTMs, serving as inputs for a CNN-based classification algorithm. The trained model achieved 100% accuracy after 1400 iterations.

Beyond using RDMs and RTMs as the only input for HGR, some research opted using spectrograms, more specifically, as the single input to the classification algorithm. For instance, in a publication from 2017, Dekker et al. [40] devised an HGR system capable of classifying three different hand gestures solely based on spectrogram features. In this study, 1000 spectrograms were recorded for each of the three different hand gestures, using a radar operating in the 24GHz band with a bandwidth of 250MHz and one receive channel. A CNN model was employed as the classification algorithm, attaining an average classification accuracy of 99% on the testing dataset. The study concluded that, in the context of these three hand gestures, there was no significant advantage in incorporating phase information from the spectrogram.

Furthermore, in 2021, Jiang et al. [41] introduced a novel HGR system designed for recognising six distinct hand gestures, again relying solely on spectrograms (DTM) as the exclusive input to the classification algorithm. An mmWave FMCW radar, specifically the AWR1642BOOST radar, was utilised, operating in the 77GHz frequency band with a 4GHz available bandwidth and equipped with 2 transmitter and 4 receiver antennas. The study investigated the impact of varying hand angles (0°, 15°, and 30°) and distances (20cm, 50cm, and 100cm) from the radar on classification accuracy. The dataset comprised of six different hand gestures performed by five participants, with 60 repetitions, resulting in a total sample size of 1800 data points. The dataset from the first four participants were employed for model training, while the dataset from the fifth participant was used for testing. Two classification algorithms, SVM and CNN, were compared. The study concluded that the highest average accuracy, 95.2%, was achieved when using CNN, with the hand positioned 20cm from the radar at a 0° angle, while SVM yielded an accuracy of only 91.4% under those conditions.

2.3.2 Multi-Channel Algorithm

Some researchers have adopted a multi-channel approach for the classification algorithms. This approach involves inputting different maps or features separately into the classification algorithm to potentially enhance accuracy. Wang et al. [42] explored a different type of convolutional neural network for HGR using the AWR1642BOOST chip FMCW radar sensor. The dataset included 10 hand gestures, totalling 4000 samples. The TS-I3D end-to-end network organised RDMs features into RTMs and DTMs as inputs for an LSTM encoder algorithm. Following that, the maps were concatenated and hand gestures were classified. The proposed TS-I3D model was compared with other classification models such as CNN, 3D-CNN, Recursive 3D-CNN, LSTM, I3D, and I3D+LSTM. The TS-I3D model achieved the highest average recognition accuracy at 96.17%. Later in 2020, Wang et al. [43] introduced an innovative approach to detecting and recognising hand gestures by integrating multiple features. This study also included a comprehensive evaluation of various classification algorithms and the utilisation of multiple spectrograms. Specifically, this research combined RTMs, DTMs, and ATMs into a Fusion DTW algorithm for hand gesture classification.

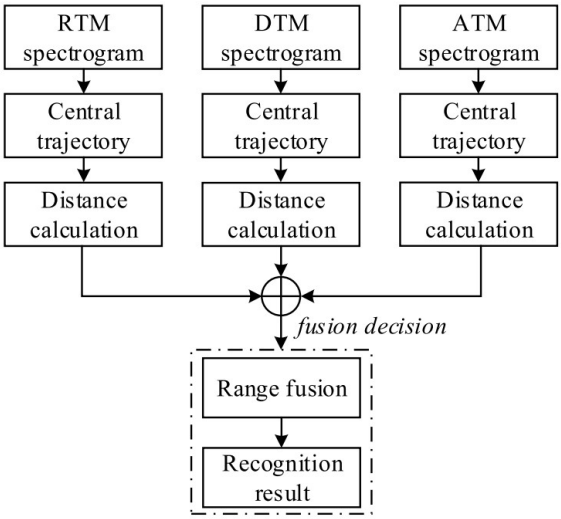


Figure 2.2: Wang et al.'s Fusion DTW algorithm with fusion of the spectrograms [45]

The mmWave FMCW radar system used was the Texas Instruments' AWR1642BOOST chip, which specifics have been previously discussed. A total of 1200 hand gesture samples from six distinct types were collected for analysis. Comparative analyses were conducted among different fusion algorithm combinations and different classification algorithms, revealing that fusing all three maps using the proposed Fusion DTW algorithm achieved the best average accuracy of 95.83%. Figure 7.1 in the Appendix highlights the methods and accuracy which were compared in the investigation.

In 2022, Arsalan et al. [44] introduced 'RadarSNN,' a resource-efficient HGR system based on the BGT60TR13C mmWave FMCW radar chip, which operates in the 60GHz frequency band and offers a 5GHz bandwidth, along with one transmit and three receive antennas. The dataset was self-constructed, comprising eight different hand gestures performed by five participants, totalling 4800 samples. Data pre-processing involved transforming raw radar data into three maps: RTMs, DTMs, and ATMs. The proposed classification model was built upon a SNN architecture, which took all three maps as separate inputs. The study compared various classification models, including CNN, 3D-CNN, MobileNetV2, and the proposed SNN model. This comparison examined both model accuracies and their respective model sizes. The top-performing model, the 3D-CNN achieved an average accuracy of 99.63%, while the SNN model delivered an accuracy of 99.50%. However, it is worth highlighting that the 3D-CNN model had a size of 12586.58 KB, while the SNN model was more compact, occupying just 75 KB.

Jin et al. [39] conducted machine learning algorithm comparison for mmWave FMCW-based HGR. A combination of RTMs and DTMs as features was employed, and a comparison of several classification models was achieved. The dataset comprised a total of 2700 data samples, collected from nine participants, encompassing six types of hand gestures with 50 repetitions each, using the AWR1642BOOST radar. The evaluated models included ID-1-D-CNN, CNN+LSTM, TS-I3D, and a proposed model based on an enhanced CNN+LSTM architecture. These models achieved average accuracies of 98.88%, 98.51%, 94.44%, and 98.18%, respectively.

Yu et al. [29] conducted a study comparing RDMs, RAMs, and a combination of both maps. The research employed the IWR1443BOOST radar, operating at 77GHz with a 4GHz bandwidth and

configured with one transmit antenna and four receive antennas. The proposed end-to-end network model combined CNN and LSTM algorithms. The dataset for this experiment comprised of 12 hand gestures performed by 12 participants with 20 repetitions for a total of 2880 samples. The classification system allowed for multi-channel input, accepting either RDM, RAM, or both. When using RDM alone, the model achieved an average classification accuracy of 81.04%, while RAM alone yielded an average accuracy of 90.51%. Notably, combining RDM and RAM improved performance further, resulting in an average accuracy of 92.74%. This investigation highlights the significance of combining multiple maps as input features for classification algorithms, as it results in an enhanced average recognition accuracy.

Similarly, Zheng et al. [46] developed the 'RGTNet,' a convolutional neural network-based HGR system that used RDMs, RAMs, and REMs as inputs. The system was designed to detect eight different hand gestures, utilising a dataset with a sample size of 5318. The IWR6843AOP radar from Texas Instruments was employed, operating at 60GHz with a 4GHz bandwidth, three transmit antennas, and four receive antennas. The 'RGTNet' network is a transformer-based framework for the classification task. The proposed network was compared to other neural networks using various input features. The best accuracy of 97.6% was achieved when combining all three maps. The experimental comparison results are presented in Figure 7.3 of the Appendix.

Yang et al. [48] delved into the integration of RTMs, DTMs, and ATMs into a multi-channel-based classification algorithm. The paper's HGR system aimed to detect and classify nine distinct hand gestures. These gestures were recorded 100 times each from five different participants, yielding a dataset of 4500 gestures. The previously discussed IWR1443BOOST radar system was employed in this experiment. A comparative assessment encompassed various classification algorithms, including LSTM, CNN-LSTM, Three-Layer LSTM, and the proposed Reused LSTM algorithm. Notably, the best performance was achieved using the Reused LSTM, with an average recognition accuracy of 98%.

In 2022, Wang et al. conducted another experiment explored the fusion of RDM and RAM. This HGR system developed by Wang et al. et al. [45] utilised the AWR1642BOOST radar and introduced a network model consisting of a dual-channel 3D-CNN with a LSTM encoder, enabling the fusion of the features and global feature extraction. The study investigated the impact on recognition accuracy when using the maps individually or a fusion of both. The system aimed to recognise eight hand gestures performed by 5 participants, for a total of 4000 samples. Results indicated that using RDM only achieved an average accuracy of 72.16% with a 3D-CNN, while RAM alone achieved 82.79% using the same architecture. However, the fusion of RDM and RAM with the dual-channel 3D-CNN improved accuracy to 86.95%. The highest accuracy of 93.12% was achieved using the 3D-CNN and LSTM encoder with both maps as inputs.

2.3.3 Multi-dimensional Features

In June 2022, Zhao et al. [49] introduced an innovative HGR approach using Texas Instrument's AWR1642BOOST mmWave radar, which operates at a frequency of 77GHz with an available bandwidth of 4GHz. The AWR1642BOOST model is equipped with two transmit and four receive antennas. The unique aspect of this implementation lies in the stitching of maps to create a unified image featuring combined features. The composite image consists of a RTM, a DTM, and an ATM.

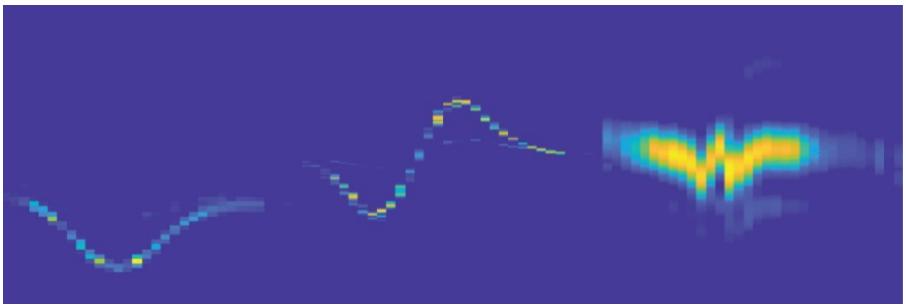


Figure 2.3: Example of Zhao et al.'s combined image

Regarding the datasets, 250 data samples were collected from two distinct participants, covering three different hand gestures, resulting in a total of 1500 data samples. Various classification algorithms were explored in this experiment, with the XGBoost classifier achieving an impressive average accuracy of 98.93%, while the Random Forest Classifier achieved a perfect 100% accuracy on the same test set.

Later in December 2022, Zhao et al. [47] published another study, this time focused on comparing the performance of systems based on single features with those relying on feature combinations. The system continued to utilise the same radar and the same three maps. A SVM served as the classification algorithm for the entire investigation. The results of this experiment concluded that the optimal performance was achieved by utilising multiple features and combining them. When using a single feature, the average recognition accuracy stood at 70.69%, 87.59%, and 69.28% for RTM, DTM, and ATM, respectively. In contrast, the combined feature encompassing all three maps yielded an accuracy of 98.48%.

In 2021, Dong et al. [51] conducted a study comparing the performance of an HGR system that employed either a multi-channel algorithm or a stitching process to generate multi-dimensional features, subsequently utilised in a single-channel classification model. The AWR1642BOOST radar system was employed for this experiment. A dataset comprising 19,760 data samples encompassing 16 different gestures, performed by 19 distinct participants, was employed for model training and testing. The evaluated models included CNN, 3D-CNN, 3D-CNN+LSTM, S3D, S3D+STDC, S3D+ASTCAC, and S3D+STDC+ASTCAC. The comparison results are presented in the Table 7.2 in the (Appendix), with the most accurate model being attained using 5D-feature cubes in conjunction with the S3D+STDC+ASTCAC approach, achieving an accuracy rate of 99.53%.

Similarly, Shen et al. [52] introduced a meta-learning HGR system. The hardware used is the previously discussed IWR1642BOOST from Texas Instruments. This system utilised five distinct maps to generate a multi-dimensional feature cube, comprising RDMs, RTMs, DTMs, ATMs, and Doppler-angle Map (DAM)s. The dataset encompassed 10 different hand gestures, each represented by 150 samples from five different participants. A 3D-CNN-based dual-channel fusion network was utilised to capture and classify hand gesture features. The research revealed that angle-based features, coupled with an increase in the number of features, led to improvements in the system's average recognition accuracy. Furthermore, the study included a comparison with more complex machine learning algorithms and feature extraction methods. The most outstanding performance was achieved using the proposed ML-HGR-Net, a 3D-CNN-based algorithm, which achieved an average recognition accuracy of 98.4%.

In 2020, Du et al. [53] conducted a study aimed at comparing the average classification accuracy of various machine learning methods using the IWR1642BOOST radar system. The performance of KNN, SVM, CNN, and 3D-CNN using a dataset consisting of 10 distinct hand gestures and involving 10 participants was evaluated. To increase the dataset, a data augmentation technique was employed, resulting in a total of 120,000 3D-feature sample images for training and testing the classification algorithms. Within this dataset, 60% of the samples were allocated for model training, while the remaining 40% were reserved for testing. The creation of 3D images was achieved by combining angle and elevation features. Notably, the study compared the average accuracy obtained when utilising a single feature (RDMs) with the 3D images. The results indicated that single-feature usage yielded an average accuracy of 91.34%, whereas the incorporation of angle and elevation features into 3D images significantly improved performance, achieving an average accuracy of 96.61%. Furthermore, the study investigated the impact of the data augmentation process. The findings revealed that the training set accuracy was higher without data augmentation, standing at 99.56%, compared to 98.89% with data augmentation. Conversely, for the testing set, data augmentation yielded better results, with an average accuracy of 96.61% compared to 94.97% without it. Lastly, the investigation introduced an attention module to the 3D-CNN model, resulting in an average recognition accuracy of 97.17% for the test set.

As previously discussed, it can be challenging to clearly discern the similarities and differences among the current state-of-the-art works. Table 2.1 below serves as a tool to consolidate the findings from each research papers, enabling a comprehensive summary of their key aspects.

Radar Model	Frequency Band	Bandwidth	Antennas	Number of Gestures/Participants/Samples	Features	Classification Algoritm	Accuracy	Reference
Soli Radar	60GHz	7GHz	2TX,4RX	11/10/2750	RDM	Random Forest Classifier	Not Disclosed	[21]
Self-Constructed	24 GHz	250MHz	1TX,4RX	7/2/2800	PRDM	LSTM	91.00%	[54]
BGT60TR24	60GHz	4GHz	1TX,4RX	5/10/2100	RDM	ConvNet +LSTM CNN+LSTM	94.34% 84.00%	[35]
Soli Radar	60GHz	6GHz	2TX,4RX	10/10/4000	RDM	RNN 2D-CNN LSTM	95.39% 92.90% 99.10%	[28]
AWR1642BOOST	77GHz	4GHz	2TX,4RX	6/4/1200	RDM	3D-CNN CNN+LSTM	95.00% 97.00%	[37]
BGT24MTR12	24GHz	4GHz	1TX,2RX	8/5/3200	RTM, CTC algorithm	3D-CNN+LSTM	96.0%	[27]
IWR6843	60GHz	4GHz	3TX,4RX	2/1/1000	RTM	CNN	100%	[38]
Self-Constructed	24GHz	250MHz	1TX,1RX	3/1/3000	DTM	CNN	99.00%	[40]
AWR1642BOOST	77GHz	4GHz	2TX,4RX	6/5/1800	DTM	SVM CNN	91.40% 95.20%	[41]
AWR1642BOOST	77GHz	4GHz	2TX,4RX	10/1/4000	RTM+DTM	CNN 3D-CNN Recursive 3D-CNN LSTM I3D I3D+LSTM TS-I3D	82.77% 88.07% 93.09% 90.35% 89.37% 93.05% 96.17%	[42]
AWR1642BOOST	77GHz	4GHz	2TX,4RX	6/1/1200	RTM+DTM+ATM	FDTW	95.83%	[43]
BGT60TR13C	60GHz	5GHz	1TX,3RX	8/5/4800	RTM+DTM+ATM	3D-CNN 2D-CNN MobileNetV2 SNN	99.63% 86.25% 99.00% 99.50%	[44]
AWR1642BOOST	77GHz	4GHz	2TX,4RX	6/9/2700	RTM+DTM	ID-1-D-CNN CNN+LSTM TS-I3D Enhanced CNN+LSTM	98.88% 98.51% 94.44% 98.18%	[39]
IWR1443BOOST	77GHz	4GHz	1TX,4RX	12/12/2880	RDM RAM RDM+RAM	CNN+LSTM	81.04% 90.51% 92.74%	[29]
IWR6843AOP	60GHz	4GHz	3TX,4RX	8/1/5318	RDM+RAM+REM	CNN CNN+LSTM 3D-CNN Proposed RGTNet	92.50% 95.70% 95.20% 97.60%	[46]
IWR1443BOOST	77GHz	4GHz	1TX,4RX	9/5/4500	RTM+DTM+ATM	LSTM CNN+LSTM Three-Layer LSTM RLSTM	72.50% 80.10% 95.30% 98.00%	[48]
AWR1642BOOST	77GHz	4GHz	2TX,4RX	8/5/4000	RDM RAM RDM+RAM RDM+RAM	3D-CNN 3D-CNN 3D-CNN 3D-CNN+LSTM	72.16% 82.79% 86.95% 93.12%	[45]
AWR1642BOOST	77GHz	4GHz	2TX,4RX	3/2/1500	Combined Image of (RTM+DTM+ATM)	XGBoost Classifier Random Forest Classifier	98.93% 100%	[49]
AWR1642BOOST	77GHz	4GHz	2TX,4RX	6/2/1500	RTM DTM RAM RTM+DTM+RAM	SVM	70.69% 87.59% 69.28% 98.48%	[47]
AWR1642BOOST	77GHz	4GHz	2TX,4RX	16/19/19760	5D-Feature Cube	S3D+STDC+ASTCAC	99.53%	[51]
IWR1642BOOST	77GHz	4GHz	2TX,4RX	10/5/7500	Feature Cube	3D-CNN	98.40%	[52]
IWR1642BOOST	77GHz	4GHz	2TX,4RX	10/10/120000	3D-Feature Image	KNN SVM CNN 3D-CNN 3D-CNN+Attention module	88.93% 90.21% 91.34% 96.61% 97.17%	[53]

Table 2.1: Table summarising the relevant research of HGR using mmWave FMCW radar in the current state-of-the-art

2.4 Current Applications

The following section delves into the contemporary applications of HGR technology. It begins by exploring applications exclusively utilising mmWave FMCW radar, such as HGR for human-vehicle interactions, intuitive device control, and HCI through cursor manipulation. Subsequently, the section broadens its scope to include HGR applications beyond mmWave FMCW radar. These applications include sign language translation, remote control of smartphones, and the control of mobile robots.

HGR represents a highly valuable tool for facilitating HCI, offering effortless communication between users and machines. Its versatility has led to its implementation in various applications, allowing users to interact intuitively with technology. Numerous instances of hand gesture interaction have been successfully deployed in the past.

Smith et al. [55] introduced a human-car interface system designed to enhance driver safety and reduce distractions. This system utilises a 60 GHz mmWave FMCW radar for HGR, enabling users to control various functions. Real-time recognition is achieved through a Random Forest Classifier algorithm. The implemented hand gestures include a 'wiggle phone sign' to initiate or end phone calls, as well as finger wiggling to pause/play music and finger counting to select playlists. Smith et al. demonstrated the ease and utility of this hand gesture interface for both drivers and passengers without causing distraction.

HGR also finds application in controlling everyday devices more efficiently. In 2018, Nguyen et al. [56] developed a wireless remote control system for computers and smart televisions using hand gestures. A 77GHz mmWave FMCW radar was employed, enabling gestures such as swiping left, swiping right, zooming in, and zooming out within a range of 0.3m to 1.2m.

Furthermore, in 2020, Li et al. [57] introduced a novel interaction paradigm called 'ThuMouse', allowing users to control a cursor with finger movements. This system utilised the IWR6843, designed by Texas Instruments, operating in the 60GHz frequency band. By employing an end-to-end classification method that combined You Only Look Once (YOLO), CNN, and RNN, Li et al. achieved real-time tracking of hand gestures using mmWave FMCW radar technology.

It's worth noting that HGR applications extend beyond mmWave FMCW radar technology. Several previous works have demonstrated HGR's potential in HCI, highlighting its broad applicability.

One early and impactful application of HGR is the translation of sign language for accessibility to both individuals and machines. In 2014, Abid et al. [58] proposed dynamic sign language recognition for smart home interactive applications. The research used image processing and a stochastic linear formal grammar (SLFG) module for HGR. The objective was to create a system enabling communication between humans and robots, especially for individuals with speech disorders or hearing impairments who rely on dynamic hand gestures as their primary mode of communication.

Another noteworthy application from 2014 is the remote control of smartphone platforms, as demonstrated by Qifan et al. [59] in a work called 'Dolphin.' This project employed ultrasonic-based technology to detect hand gestures, combined with machine learning classification methods for accurate recognition. Similar to mmWave FMCW radar technology, this system extracted features from Doppler shifts to identify predefined gestures. It achieved real-time detection of 24 different gestures with an average

accuracy of 93% in two typical environmental settings.

Lastly, in 2019, Simão et al. [36] developed an application for communication and remote control of mobile robots. Their system utilised wearable sensor technology for HGR, enhancing the accessibility and controllability of collaborative robots. Classification models achieved an average accuracy of 95.6% for 24 static gestures using a Random Forest Classifier and an average accuracy of 99.3% for 10 dynamic gestures using an artificial neural network. Simão et al. successfully employed this HGR system to remotely control a robot in preparing a breakfast meal.

2.5 General System Limitations

Despite the numerous advantages offered by mmWave FMCW radar for HGR, it is evident that several technical challenges can influence the final result of the system. This section delves into these significant limitations, starting with the challenges related to real-time recognition requirements. Subsequently, the constraints associated with environmental conditions are discussed before addressing how the various types of gestures can affect the HGR system.

2.5.1 Real-Time HGR

Currently, HGR implementations achieve high accuracy owing to their large data samples and complex classification models. Wang et al. [60] have noted significant constraints associated with large data sample sizes. Increasing data sample sizes lead to more complex neural network models, resulting in escalating training costs. Complex classification models, as demonstrated by previous studies [44, 51], demand extensive memory and computational resources. This poses a challenge for potential commercial applications, given that most embedded systems, have a restrained available memory and processing power [22]. Consequently, complex classification models for HGR often struggle to meet the real-time application requirements. While some research endeavours aim to implement lightweight networks to reduce model depth and training times, simplifying the model's complexity while maintaining high average accuracy in HGR remains a challenge.

2.5.2 Environmental Conditions of the System

In the current state-of-the-art, most implementations assume ideal experimental scenarios where interferences, such as noise, have minimal impact. Past research, such as the study conducted by Zhang et al. [27] in 2018, frequently necessitates that participants remain stationary during experiments. However, practical applications rarely offer ideal environmental conditions, and interferences significantly affect the performance of HGR systems. Despite some research efforts, like the investigation by Rodrigues and Li [30] , which was discussed earlier, there has only been surface-level exploration of the applicability of mmWave FMCW technology under realistic conditions.

2.5.3 Limitations due to Gesture Types

Lastly, significant limitations arise concerning the type of gestures when employing mmWave FMCW radar for HGR. In the current state-of-the-art, previous research typically relies on larger gestures due to their richer movement data, facilitating feature extraction. Wang et al. emphasise this aspect

by noting that the radar's bandwidth limitations result in reduced average accuracy as the gesture's activity range decreases. This is evident in a study conducted by Zhang et al. [61] in 2019, where different gesture scales were compared. Gesture sizes were set at 0.5m, 0.3m, and 0.1m, yielding average accuracies of 95%, 87%, and 76%, respectively. This study demonstrate how gesture size and type can profoundly impact the system's performance. Systems requiring recognition of smaller-scale hand gestures typically demand more from HGR, leading to increased complexity, memory usage, and computational requirements.

2.6 Chapter 2 Summary

This chapter presents a comprehensive literature review focused on HGR using mmWave FMCW radar. Initially, it introduces the concept of HGR by discussing its history, starting with the first implementation of an HGR system. Subsequently, it delves into various approaches to HGR and traces the evolution of the field, highlighting mmWave FMCW radar as one of the most effective methods. It also introduces the Google Soli project [31], the first implementation of mmWave FMCW radar in HGR.

The past implementations of HGR using FMCW radar are then examined through the discussion of the three stages in the system's process. In this section, the first stage, involving how to get the raw radar data, is introduced. Past research findings concerning the types of gestures, the datasets employed, and the different radar models features, are outlined. Subsequently, the second stage, involving the processing of raw radar data, is introduced; and a discussion of the diverse techniques employed in previous research are discussed. Lastly, the third and final stage is discussed, highlighting the array of feature extraction and classification algorithms employed in earlier works.

Additionally, this literature review incorporates a section dedicated to related works, which reviews all the relevant past implementations of HGR using mmWave FMCW radar. Current applications of HGR are subsequently discussed, providing insights into the current utility of this technology. Finally, the chapter concludes by addressing the overarching limitations of HGR using mmWave radar, including challenges associated with real-time requirements, environmental conditions, and the types hand gestures.

This comprehensive literature review forms the foundation for the rest of the report, facilitating the selection of the different components that will influence the design of the HGR system.

Chapter 3

Theoretical Background

This chapter is dedicated to exploring the essential theoretical background necessary for understanding the project. The chapter commences by introducing radar technology and establishing a foundation in general radar theory. It then delves into the theory behind FMCW radar, exploring the equations that are relevant to the project. Subsequently, the chapter addresses the theoretical aspects of raw radar data processing and the methods for generating various maps. The chapter concludes with a discussion of CNN theory.

3.1 General Radar Theory

3.1.1 Basic Radar Fundamentals

Radar, which stands for 'Radio Detection and Ranging', is an EM system employed for detecting and localising objects or individuals within a specific area [62]. It functions by emitting EM energy from an antenna, which is intercepted and reflected by objects, commonly referred to as 'targets' [62]. The process of detection determines the presence or absence of a target, while localisation assesses the range, or distance, between the radar and the target. Radars find utility in various technologies primarily due to their capability to detect and locate targets under conditions where human vision is inadequate or impossible [62].

A typical radar system comprises of several key components, namely a transmitter, an antenna, and a receiver, all interconnected [63]. The transmitter generates high-frequency EM waves, or Radio Frequency (RF) signals. The radar's antenna then emits the radar signal generated by the transmitter into the environment in a specific direction, also collecting the reflected signals from the target [62]. The receiver of the radar subsequently amplifies the reflected signal, enabling the extraction of information about the target. Figure 3.1 below provides a simplified illustration of this process.

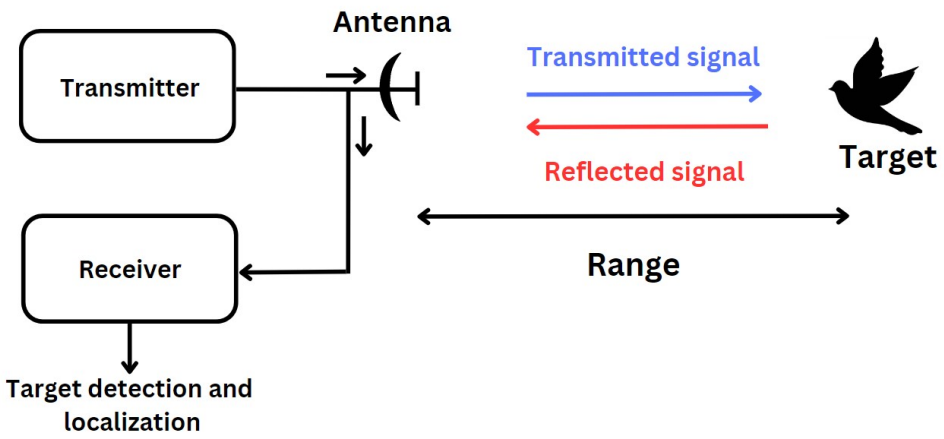
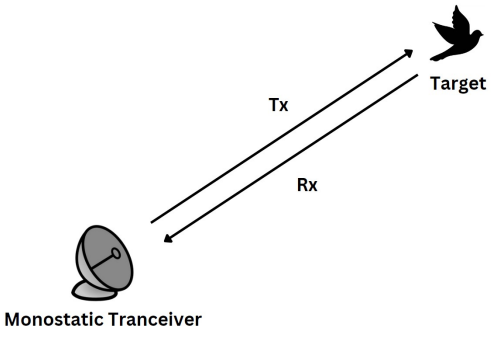
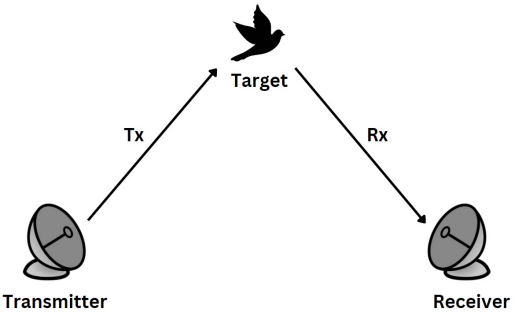


Figure 3.1: Basic Radar configuration

In radar system design, the number and positioning of transmitters and receivers are referred to as the radar geometrics [63]. There exist two fundamental radar system types: monostatic radar and bistatic radar. Monostatic radars employ the same antenna for both transmitting and receiving signals, while bistatic radars use distinct antennas for these two functions [63]. These radar types are exemplified in Figure 3.2 below.



(a) Monostatic Configuration



(b) Bistatic Configuration

Figure 3.2: Types of radar geometrics

3.1.2 Continuous-Wave radar

Continuous Wave (CW) radar systems continuously emit EM radiation while simultaneously receiving reflected signals from targets. A notable characteristic of CW radars is that when there is no target or a stationary target, the frequency of the transmitted signal remains unchanged [63]. However, as soon as a target is in motion within the radar's range, the reflected signal experiences alteration due to the Doppler effect [63]. This altered frequency, known as the Doppler frequency, can be analysed to determine the motion of the target. The Doppler shift signifies the change in frequency of the EM signal caused by the motion of the target, the transmitter, or both. Consequently, CW radars are capable of measuring the instantaneous rate of change in the range of the target. An important limitation of CW radars is that the transmitter is not modulated, making them incapable of providing information about the target's range [63].

3.2 FMCW Radar

The measuring of the range of a target in CW radar systems necessitates a precise timing reference encoded into the transmitted waveform. This reference is established through the incorporation of frequency modulation into the radar system, resulting in what is commonly known as FMCW radar [63]. The modulation of the transmitted signal can take various forms, with sawtooth linear waveforms being the most common.

3.2.1 Signal Description

Figure 3.3 below illustrates a sawtooth linear FMCW waveform that represents the transmitted and received frequency signals in an FMCW radar configuration as a function of time [64].

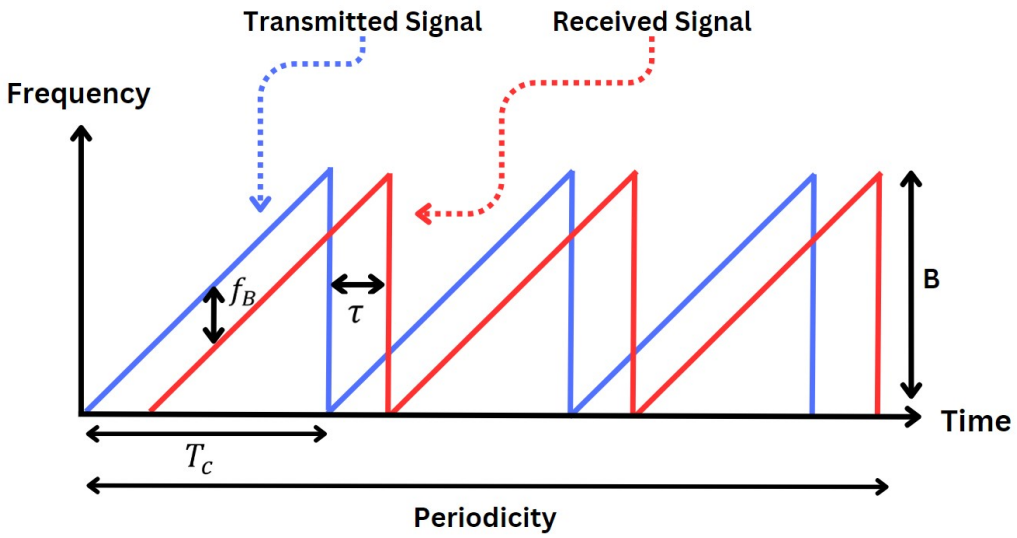


Figure 3.3: Transmit and receive frequency as a function of time

The transmitted signal and reflected signal can be represented by the following equations [50]:

$$s(t) = \exp\left(j2\pi\left(f_c t + \frac{Bt^2}{2T_c}\right)\right) \tag{3.1}$$

$$r(t) = \exp\left(j2\pi\left(f_c(t-\tau) + \frac{B(t-\tau)^2}{2T_c}\right)\right) \tag{3.2}$$

where,

f_c = Carrier (modulated) frequency

τ = Time delay

B = Modulation bandwidth

T_c = Chirp time

The resulting beat frequency, labelled as 'f_B' in Figure 3.3, represents the instantaneous difference in

frequency between the transmitted and received signals. Utilising the beat frequency, the determination of the target's range, speed, and angle information can be done [63]. The chirp time 'T_c' represents the time period of the signals, signifying the duration between repetitions of the sawtooth waveform. Each frame of data comprises multiple chirps concatenated over time, known as the frame periodicity [64].

The time delay for echo (received signal), denoted as 'τ', corresponds to the round-trip propagation time between the transmit and receive signals [64]. The modulation bandwidth 'B' signifies the total peak-to-peak frequency deviation.

The beat frequency can be expressed using these parameters in the following equation, known as the FMCW equation:

$$f_B = \frac{2R}{c} \cdot \frac{B}{T_c} \, [Hz] \tag{3.3}$$

where,

R = range (distance between radar and target)

c = speed of light = propagation velocity = 3.0×10^8 m/s

Consequently, the target's range 'R', in FMCW radar can be determined by rearranging the FMCW equation as follows [65]:

$$R = f_B \cdot \frac{c \cdot T_c}{2B} \, [m] \tag{3.4}$$

Target velocity can be determined using the Doppler effect [64], which accounts for the Doppler shift resulting from the motion of the target. As discussed earlier, the beat frequency signifies the difference between the transmitted and received signal frequencies [63]. The maximum target velocity 'v_{max}' can be determined using the wavelength 'λ' and chirp time 'T_c' of the signal:

$$\lambda = \frac{c}{f_c} \, [m] \tag{3.5}$$

$$v_{max} = \frac{\lambda}{4T_c} \, [\text{m/s}] \tag{3.6}$$

Resolution in radar systems is a vital aspect that directly influences radar performance [65]. Range resolution refers to the radar's ability to distinguish targets in more detail [66]. This parameter can be determined using the bandwidth of the transmitted signal and the speed of light as follows [65]:

$$R_{res} = \frac{c}{2B} \, [m] \tag{3.7}$$

Velocity resolution, on the other hand, refers to the radar system's ability to distinguish relative velocities of different targets [65]. A smaller velocity resolution is preferred, as it enables the detection

of smaller changes in target velocity. Velocity resolution is influenced by the duration of a data frame and the number of chirps within it. The velocity resolution can be described as:

$$v_{res} = \frac{\lambda}{2 \cdot N \cdot T_c} \ [m] \tag{3.8}$$

Enhancing the range resolution can be achieved by increasing the bandwidth of the transmitted signal. While increasing the number of chirps in a data frame, will improve the velocity resolution. These improvements lead to superior radar performance in distinguishing targets and their velocity characteristics [64].

3.2.2 Raw Radar Data

Before extracting valuable information about a target, received raw radar signals undergo several essential processing steps, as described by Wang et al. [60]. These steps are illustrated in Figure 3.4 below.

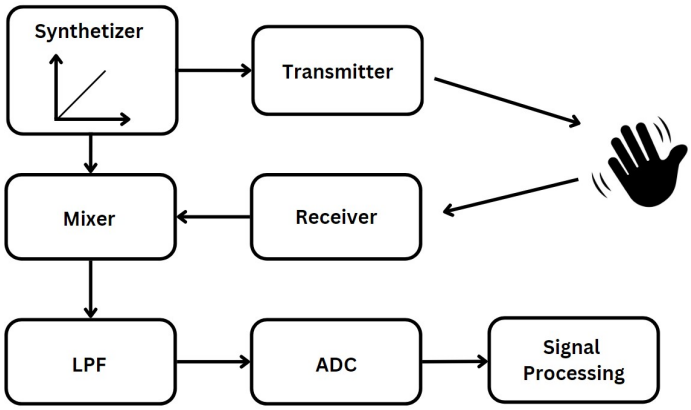


Figure 3.4: Block Diagram of a basic FMCW radar data acquisition

Initially, the FMCW radar generates a linear frequency-modulated waveform using a synthesizer. The transmitter then emits this frequency-modulated signal, which is intercepted and reflected by the target. The mixer combines the transmitted and received signals, producing a beat frequency. This beat frequency is further processed by a low-pass filter and an analogue-to-digital converter for analysis. The signal processing can take place either within an onboard digital processing chip or on an external computer [60]. Subsequent to data acquisition stage, the raw radar data needs to be processed to extract information about the target's range, velocity, and angle.

3.3 Radar Processing

3.3.1 Pre-processing

After the radar data is extracted, it initially takes the form of a complex vector, organised into frames of data. Each frame represents a specific time interval and contains pertinent radar data for that duration [64], as illustrated in Figure 3.5 below.

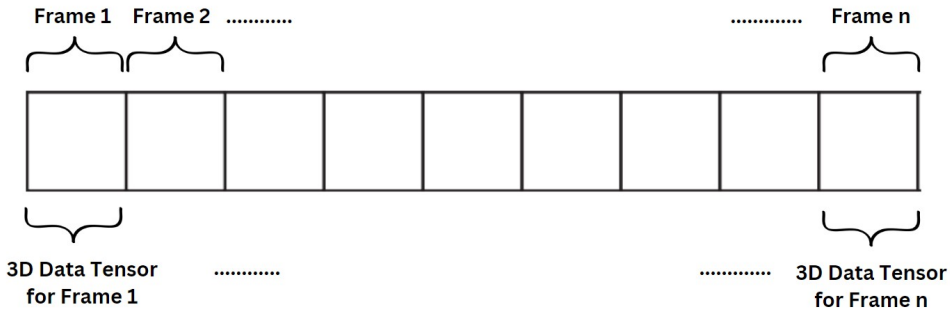


Figure 3.5: Visualisation of frame vector where each frame contains a 3D data tensor

Each frame is composed of a 3D tensor that encapsulates the data. This 3D data tensor encompasses the samples for each chirp across all channels [64]. The structure of this data tensor is depicted in Figure 3.6 below.

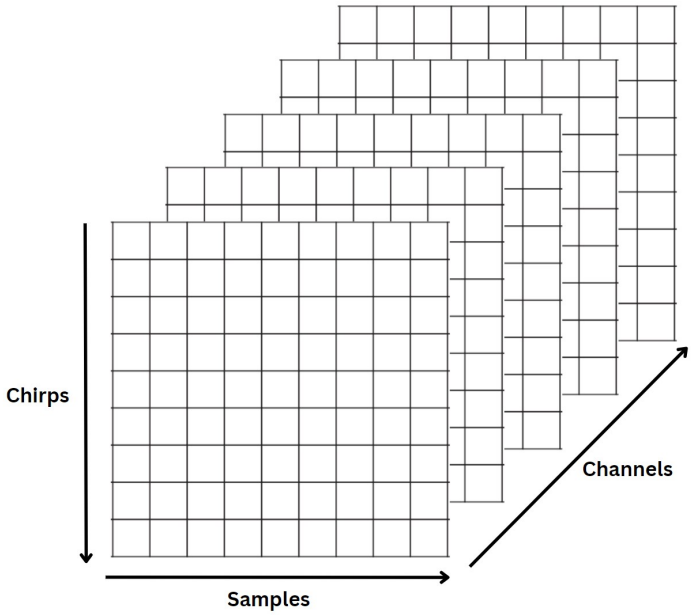


Figure 3.6: 3D Data Tensor for a single frame

In the context of FMCW radars, there are multiple transmitting and receiving antennas, which results in the presence of several channels. The total number of channels is determined by multiplying the number of transmit antennas with the number of receive antennas. Each channel contains data concerning samples and chirps, as evident from Figure 3.6.

To create visual representations and analyse data, processing is typically performed on one channel [64]. This is achieved by selecting a specific channel or by summing data from multiple channels, thus converting the data into a two-dimensional matrix (for one frame). The matrix rows represent the fast time domain, which contains samples for one chirp, while the columns correspond to the slow time domain and contains the different chirps, as depicted in Figure 3.7 below.

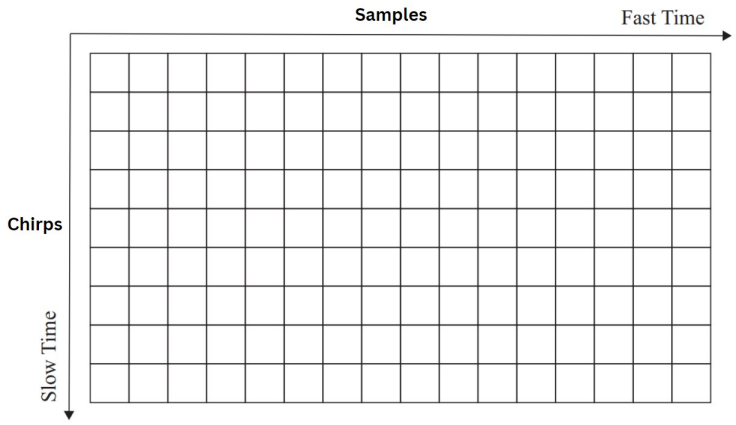


Figure 3.7: Fast Time and Slow Time Matrix [64]

3.3.2 Range-Time Maps

RTM are invaluable tools for FMCW radar data processing. These two-dimensional plots provide insights into a target's range over a defined time period, facilitating the analysis of the target's movement, direction, or its stationary state. Analysing the range data over time is essential because target's actions/movements typically occur over a duration rather than instantaneously. An example of an RTM is displayed in Figure 3.8 below.

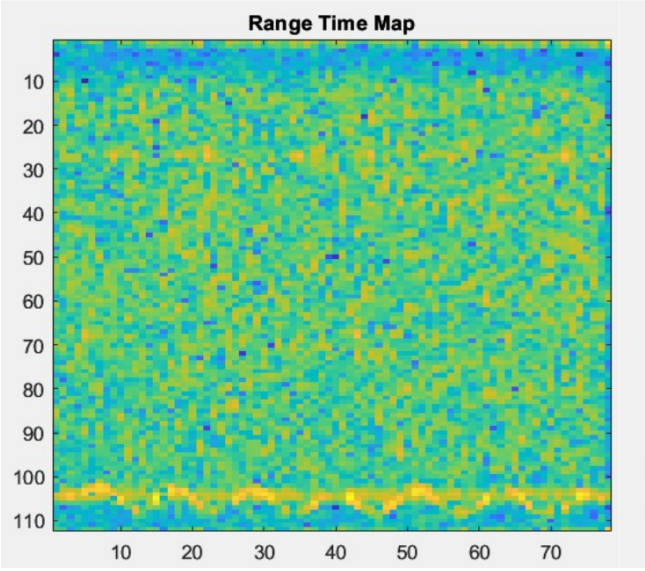


Figure 3.8: Example of a RTM

To create an RTM, a Fast Fourier Transform (FFT) is performed in the fast-time domain (sample index) [64]. This FFT operation makes the target's range information accessible within this domain. In the specific context of generating RTMs, the analysis focuses on a single chirp of data, resulting in the generation of a vector. This vector is subsequently plotted over a specified period of time, typically represented in frames.

3.3.3 Range-Doppler Maps

RDMs, also known as Range-Doppler Images, provide a two-dimensional representation of a target's range and velocity at a specific point in time (frame). An example of an RDM is depicted in Figure 3.9 below.

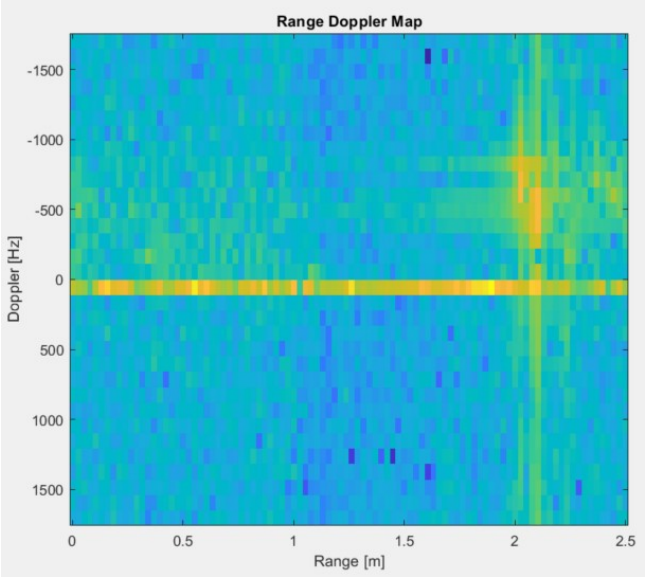


Figure 3.9: Example of a RDM with a target at 2.1m

RDMs are important tools for detecting, localising, and tracking targets. To generate RDMs, the data matrix containing radar data undergoes manipulation similar to that for RTMs. After performing FFT in the fast-time domain (sample index) to acquire range information, a second FFT is executed in the slow-time domain (chirp index) [64]. This yields a two dimensional matrix containing the range and Doppler (velocity) information of the target at a particular frame. RDMs are valuable for determining the range bin in which a target resides, enabling further data processing at that particular range. The process is visualised in Figure 3.10 below.

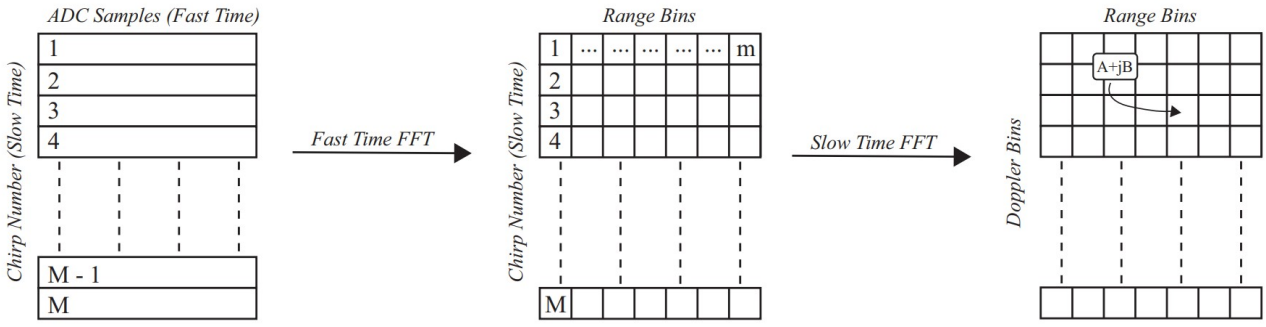


Figure 3.10: RDM process [64]

3.3.4 Doppler-Time Maps

DTMs are often represented as spectrograms and provide a visual representation of a target's velocity at a specific range over time. An example of a DTM is shown in Figure 3.11 below.

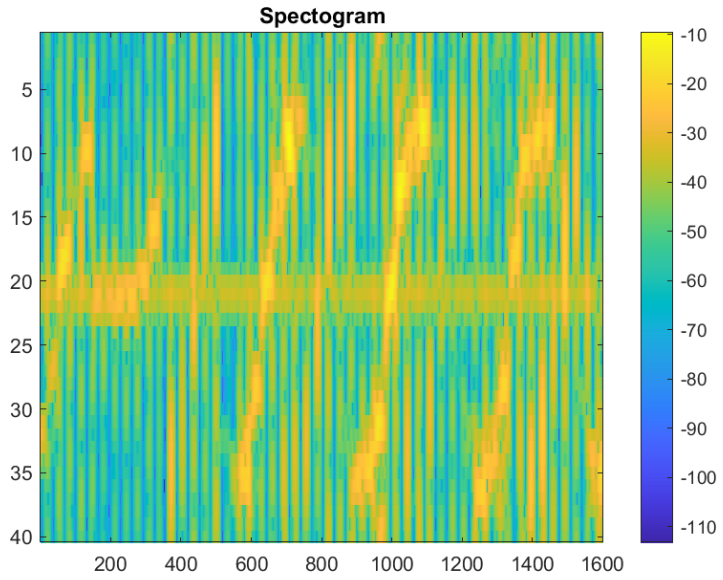


Figure 3.11: Example of a DTM

DTMs are generated using the Short Time Fourier Transform (STFT). The STFT is essentially a Fourier Transform of the signal multiplied by a window sliding across the domain over time [64]. The choice of window size and overlap plays a critical role in DTM generation. To obtain DTMs, STFT is applied to the data matrix, which has already gone through a Fast Time FFT to get the range information. The process is visualised in Figure 3.12 below.

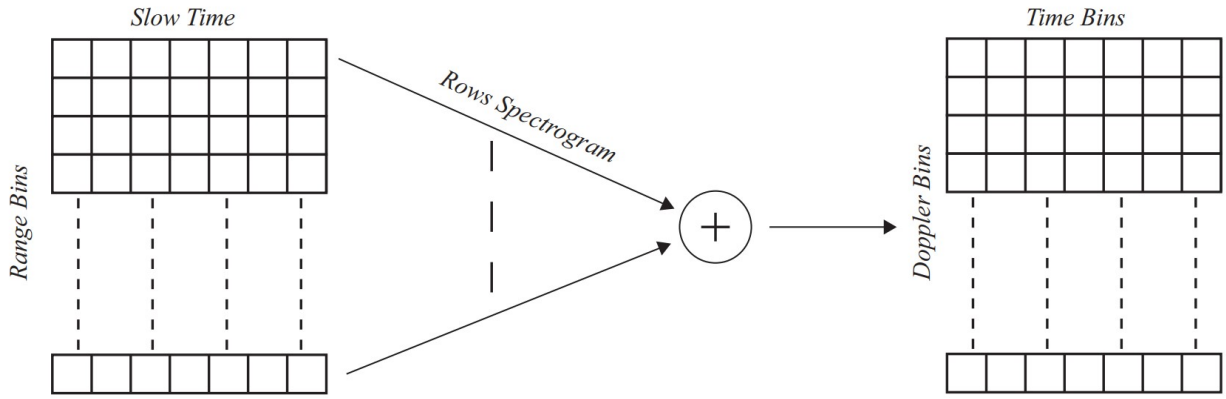


Figure 3.12: Doppler-Time Spectrogram process [64]

3.3.5 Clutter

Clutter refers to the undesirable signals that are reflected by objects present within the radar system's field of view [67]. Clutter originates from various physical elements within the natural environment, including walls, the ground, and so forth. The presence of clutter significantly affects the detection of targets within the environment, often resulting in false alarms, where objects other than the intended target are mistakenly identified [67]. To minimise the challenges posed by clutter, radar signal processing has seen the development of clutter extraction and clutter removal techniques, which

effectively mitigate the display of extraneous signals. Further information about clutter can be found in [68].

3.3.6 Windowing

Windowing is a technique that entails multiplying the data by a window function before applying the FFT. This approach provides several advantages, notably the reduction of side lobe amplitudes [69]. Side lobes refer to unintended peaks that appear on both sides of the main lobe of the signal. The mitigation of side lobes is useful as these peaks can introduce ambiguity in target detection and complicate the differentiation of multiple targets [69].

Nevertheless, it's essential to recognise that there exists a trade-off between the width of the main lobe and the level of side lobes. A broader main lobe width may have disadvantages in certain contexts. The choice of which windowing function to use, such as Hann, Hamming, or Blackman, depends on the particular requirements of the experiment and the desired results [69]. For further information about the features of windowing, see [70].

3.4 Convolutional Neural Networks

3.4.1 Fundamentals of Convolutional Neural Networks

CNNs represent a specialised category of deep neural networks primarily employed in the processing and analysis of images [71]. These networks find their most common application in classification tasks, wherein the objective is to categorise images into distinct classes. The core technique of CNNs is the mathematical operation known as 'convolution'. [72]. Convolution is a mathematical operation that combines two distinct functions to generate a third function [73].

Input images, made up of pixels, are regarded as matrices that pass through layers, undergoing various transformations. Within these matrices, individual blocks are known as neurons, functioning as the processing units of the neural network [74]. Neurons operate the same as functions, taking inputs and returning an output. However, to produce an output, a neuron must first undergo activation [74], achieved through the use of activation functions, a concept elaborated upon in Section 3.4.3.

CNNs typically consist of multiple layers designed to reduce input images into a format that facilitates feature identification within the images. This enables the network to make predictions and classify images effectively [72]. An illustration of a basic CNN architecture is presented in Figure 3.13 below.

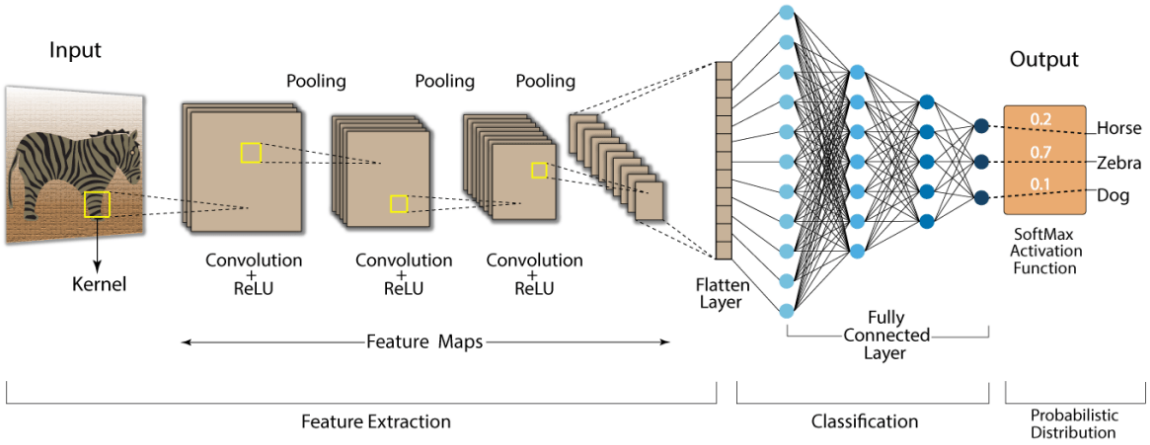


Figure 3.13: Example of a CNN model architecture [75]

3.4.2 Different Layers

Convolutional Layers

The convolutional layers are the main aspect of a CNN. These layers contain a set of filters, commonly known as kernels, which are used to extract the features of the input images [76]. These kernels have the same dimensions as the input but have a smaller size. The kernel usually slides across the rows and columns of the input images in order to create an activation map, this is the convolution mathematical operation process discussed above [76]. This convolution process can be illustrated by the following Figure 3.14.

Convolutional layers constitute the core component of CNNs. These layers contain a set of filters, often referred to as kernels, designed to extract features from input images [71]. Kernels have the same dimensions as the input but are smaller in size. This operation involves the kernel sliding across the rows and columns of the input images, performing a convolution process to create an activation map [76]. A graphical representation of this convolution operation is shown in Figure 3.14.

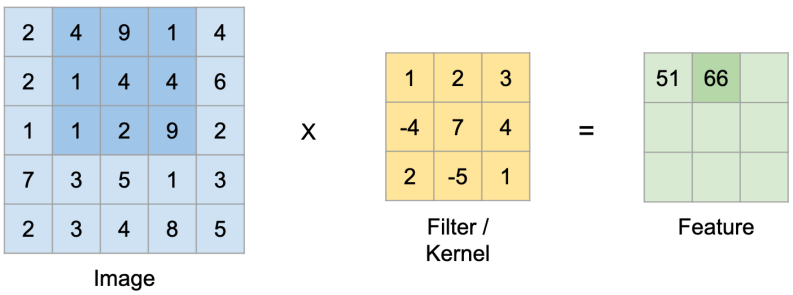


Figure 3.14: Illustration of the convolution process [75]

Batch Normalisation Layers

Batch normalisation layers are normalisation techniques applied between the various layers of CNNs. The primary purpose of these layers is to normalise the inputs at each layer during the network's training phase [77]. This normalisation is performed on mini-batches of data rather than the entire dataset, hence the name 'batch normalisation.' It enables the transformation and rescaling of data from different sources to a standard scale, facilitating their analysis, comparison, and combination [77].

Batch normalisation layers offer several advantages. They accelerate the training process and allow for the use of higher learning rates, thereby simplifying the learning process. Moreover, these layers enhance training stability by reducing the internal covariate shift, which refers to changes in data distribution [77]. This ensures that the data training process remains effective, with data sizes less likely to become excessively large or small during training. Additionally, batch normalisation layers have a regularising effect, which can help against overfitting [77], a concept explored in Section 3.4.4.

Pooling Layers

Pooling layers are often placed between each of convolutional layers in a CNN. These layers play a vital role in effectively decreasing the resolution of feature maps generated by the preceding convolutional layers [75]. They are critical in managing the dimensions of the data, which, in turn, enables CNN models to learn diverse features from the data while also reducing computational costs by decreasing the dataset size [78]. Pooling layers effectively summarises the features present in the feature maps after the convolutional layers.

Several types of pooling methods exist, including max pooling, lp pooling, and average pooling. The technique of particular interest for this project is max pooling [75]. In this method, a kernel traverses the output matrix of the preceding convolutional layer, selecting the maximum value from each window, which results in downsizing the matrix. This process is visually depicted in Figure 3.15.

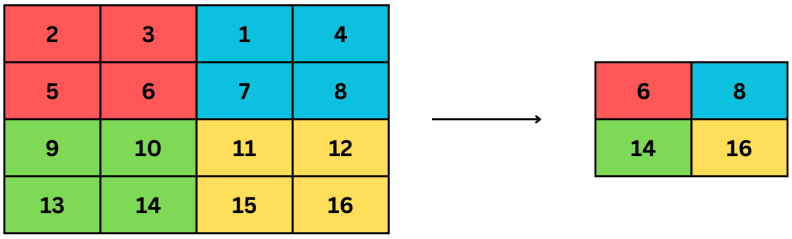


Figure 3.15: Max pooling example [78]

Flatten Layers

Flatten layers find their place towards the conclusion of the feature extraction stage of the CNN model architecture, as seen in Figure 3.13. Their primary purpose is to transform data from its existing multi-dimensional size into a one-dimensional vector, all while preserving its contents [79]. This operation serves as a preparation step for the final layer of CNN model, known as the fully connected layers. These fully connected layers require a one-dimensional vector as input, which necessitates

this transformation, given that data from the preceding layers typically isn't one-dimensional [79]. Essentially, flatten layers act as bridge between the other layers within the system and the fully connected layers.

Fully Connected Layers

In a deep CNN model, there may be multiple fully connected layers, as depicted in Figure 3.13. These fully connected layers play a crucial role in connecting all the neurons from the preceding layers [75]. They are the key components responsible for the data classification process. The ultimate fully connected layer in the CNN model serves as the final output layer and contains the predictions of the classes [75].

3.4.3 Activation Functions

Activation functions play a pivotal role in CNNs, having significant impact on the overall performance of the models [75]. These functions essentially act as 'switches' for neurons within the CNN, allowing for the outputs of the neural network to be determined [80]. Several activation functions exist, including (Rectified Linear Unit) (ReLu), Sigmoid, SoftMax, and others. For this project, only one specific activation functions is featured: ReLu.

The ReLu function has gained popularity due to its simplicity and computational efficiency compared to other activation functions [75, 80]. It is defined as:

$$R(z) = max(0, z) \tag{3.9}$$

To provide a visual understanding, Figure 3.16 below illustrates the ReLu function.

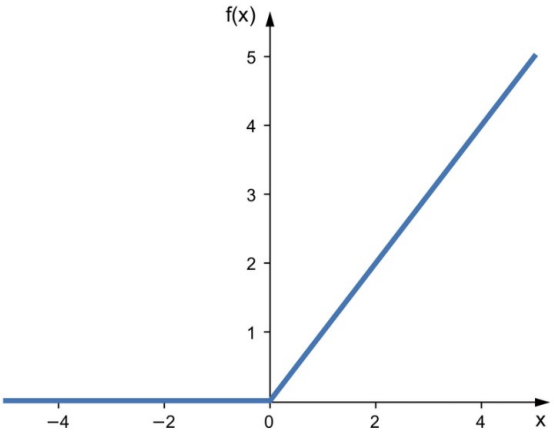


Figure 3.16: ReLu activation function [81]

3.4.4 Training Convolutional Neural Networks

Training Process

The training process for CNNs necessitates the splitting of the image dataset into three distinct subsets: the training set, the validation set, and the testing set. The allocation of these subsets is not dictated by an optimal percentage split but instead depends on the specific requirements of the experiment [82].

The process is a multi-step procedure in which the model is progressively refined. Initially, the model is trained using the training dataset. The learning process begins by inputting an image into a network of neurons, which then processes it through various layers to produce an output [83]. The highest output value corresponds to the category associated with the input image. This initial step is known as the 'feed-forward' phase. Following this, the model is evaluated using a validation dataset, and necessary adjustments are made based on the obtained results [82].

These adjustments involve fine-tuning the model's weights and biases through the utilisation of loss functions and optimisers. The 'loss' is the difference between the model's output value and the actual value [83]. The loss function incorporates all the weights and parameters of the neural network. The adjusting of the model parameters is done using a numerical optimisation algorithms. Minimising this loss is achieved by adjusting the weights and biases, which is referred to as 'backpropagation'. The number of iterations the model undergoes during training and validation is commonly referred to as 'epochs'. The entire process, including feed-forward, backpropagation, and the adjustment of model weights and biases, is illustrated in Figure 3.17 below.

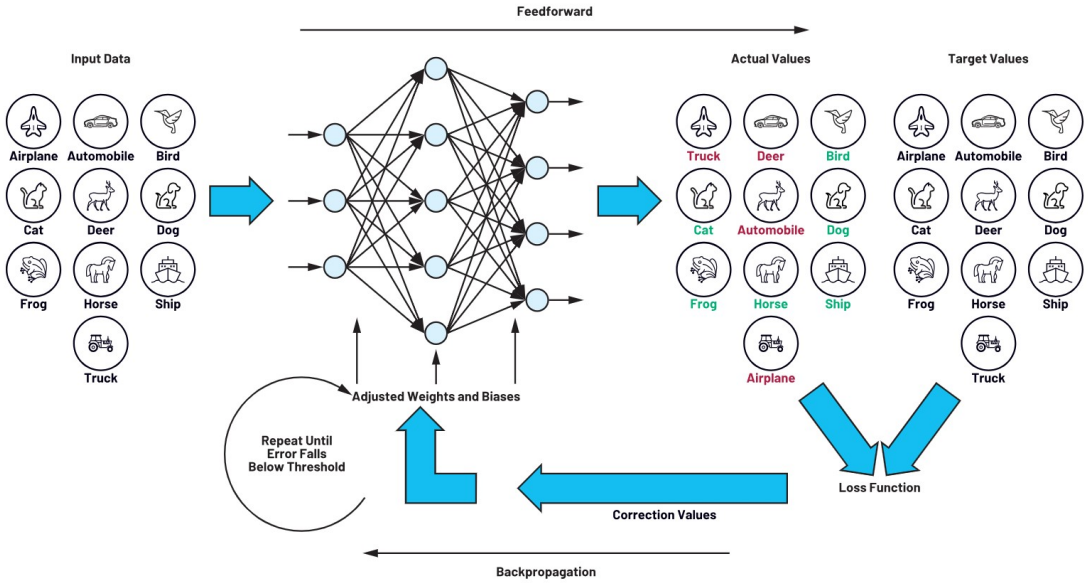


Figure 3.17: CNN Model Training Process [82]

Once the training and validation on the validation dataset are completed, the system assesses various metrics, such as validation accuracy, validation loss, and training loss. At this point, the model's hyperparameters are fine-tuned to enhance its performance. When a suitably performing model is obtained, it undergoes testing using the testing dataset [82]. Further information about the training process of CNN can be found in [83].

Hyperparameters

Hyperparameters are predetermined values that are configured before the training of the CNN. These settings remain constant and are not influenced by the data, unlike model weights and biases. The adjustment of hyperparameters usually depends on the specific requirements of the system and often necessitates multiple iterations to fine-tune and optimize the model's performance [84].

There are numerous hyperparameters that can be adjusted, including but not limited to the learning rate, learning rate decay, learning rate decay step, weight decay, and batch size. More detailed information about these parameters can be found in [84].

Generalisation

Generalisation refers to a model's performance on new, previously unseen data [85]. A CNN model possesses effective generalisation when it can make predictions on new, unseen data with a level of accuracy similar to that achieved on the training data [86]. Several techniques are available to enhance the system's generalisation, including data augmentation, regularisation, and hyperparameter tuning, among others.

Data augmentation techniques involve applying various transformations to the existing dataset to increase its size and diversity [86]. These techniques are particularly valuable when working with limited datasets and allow for the robustness of the model to be enhanced. Common data augmentation techniques include random cropping, colour space alterations, translation, reflection, flipping, rotation, and resizing of the data to generate a larger and more diverse dataset.

When the validation accuracy of a model overly exceeds the testing accuracy, it signifies that the model's generalisation is inadequate [83]. This concept is referred to as 'Overfitting'. Overfitting occurs when the model excessively tailors itself to identify the particular input images within the training dataset. As a consequence, the model becomes too complex and has too many parameters [83].

3.5 Chapter 3 Summary

This chapter has provided the essential theoretical background for the exploration of HGR using mmWave FMCW radar technology. It begins by discussing the fundamental radar principles, including an introduction to various radar system types. The discussion then introduced CW radars and delved more specifically into FMCW radar systems. The various aspects of FMCW radar systems were explored namely: signal description, data acquisition procedures, and the structure of raw radar data. Subsequently, the chapter delves into radar signal processing techniques, providing the benefits of different mapping approaches. This part of theory also provides insights into the acquisition RDMs, RTMs and DTMs and strategies for maximising the information they yield through the application of processing tools, such as windowing or clutter removal. Ultimately, the chapter concludes with theory on classification algorithms, with a particular emphasis on CNNs. This section outlined the fundamentals of CNNs, including the analysis of different layers within a CNN model, the role of activation functions, and the training process of CNN models. This theoretical background chapter established the basis for the subsequent design phase of the project.

Chapter 4

System Design

The following chapter outlines the design methodology employed for the implementation of the proposed HGR system using mmWave FMCW radar. It delves into the intricacies of the design process and the various decisions made, guided by the theoretical background laid out in Chapter 3. Throughout the design process, an agile approach was adopted, which involved breaking down the entire system into smaller subsystems. The chapter commences by providing an overview of the system as a whole and the sequential steps taken in the design process. Subsequently, it delves into the selection of specific gesture types and discusses the hardware components used in the design. The process of collecting raw radar data is then detailed, describing the dataset itself and the methodology employed for data acquisition. Subsequent sections of the chapter focus on discussing the design of the processing pipeline and the transformation of raw radar data into maps. The chapter concludes by presenting the design of the proposed classification algorithm, which will allow for the distinguishing and categorising of the various types of hand gestures.

4.1 Design Process

4.1.1 System Pipeline Overview

The system's implementation followed a sequential methodology, commencing with the design of the dataset. This process was divided into four distinct phases, labelled as A, B, C, and D in Figure 4.1, each composed of various tasks that collectively constitute the overall system design.

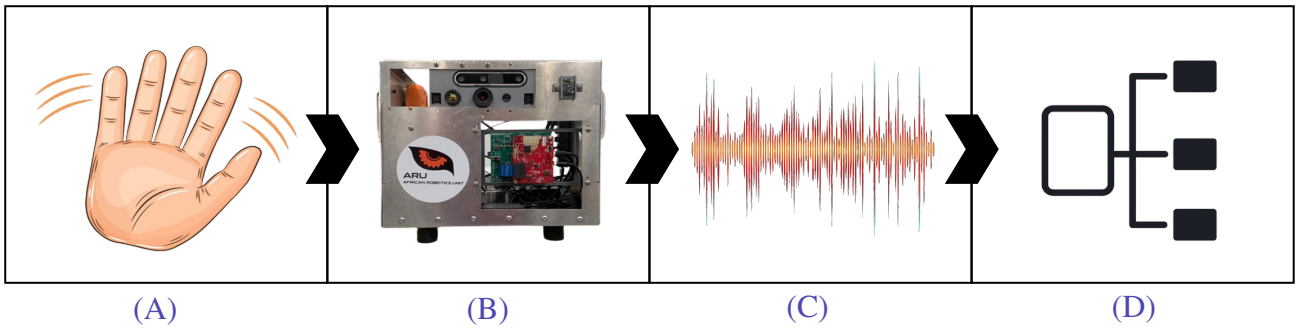


Figure 4.1: Flowchart showing the design pipeline for HGR using mmWave FMCW

The initial phase, labelled (A), primarily encompassed the definition and selection of the different hand gestures categories, the number of participants and the number of samples required for the experiment. The second design stage, labelled (B), comprised the acquisition of raw radar data. This involved the selection and justification of hardware and software choices for the project. Furthermore, this phase encompassed radar system configuration, environmental setup, parameter selection, as well as data conversion and transmission to enable the subsequent stage. The third stage, labelled (C), featured the development of the data processing pipeline. This phase, too, consisted of multiple discrete tasks, including the reading of the data, its processing and the plotting of the maps. The final stage of the design, labelled (D), involved machine learning design. This stage comprised several steps, namely, the selection of a classification algorithm, feature selection, model design, and model training.

4.2 Hand Gesture Design

4.2.1 Types of Hand-Gestures

The selection of the types of hand gestures was the result of an extensive analysis of the current state-of-the-art and the potential applications of HGR for HCI. This choice was influenced by several factors, such the consideration of micro-gestures, which primarily involve subtle finger movements and have inherent limitations.

Given that micro-gestures involve small-scale finger movements, hand gestures that predominantly involve motion or combinations of movements were chosen. This decision aligns with radar theory, discussed in Section 3.2, which suggests that identifying minor variations in target velocity or range necessitates lower resolution. Consequently, micro-gestures, due to their nature, pose challenges for effective identification, potentially impacting radar performance and the overall system.

Five distinct hand gestures were implemented for the project, including grabbing (a), lifting (b), pulling (c), pushing (d) and patting (e). These gestures are visually depicted in Figure 4.2 below.

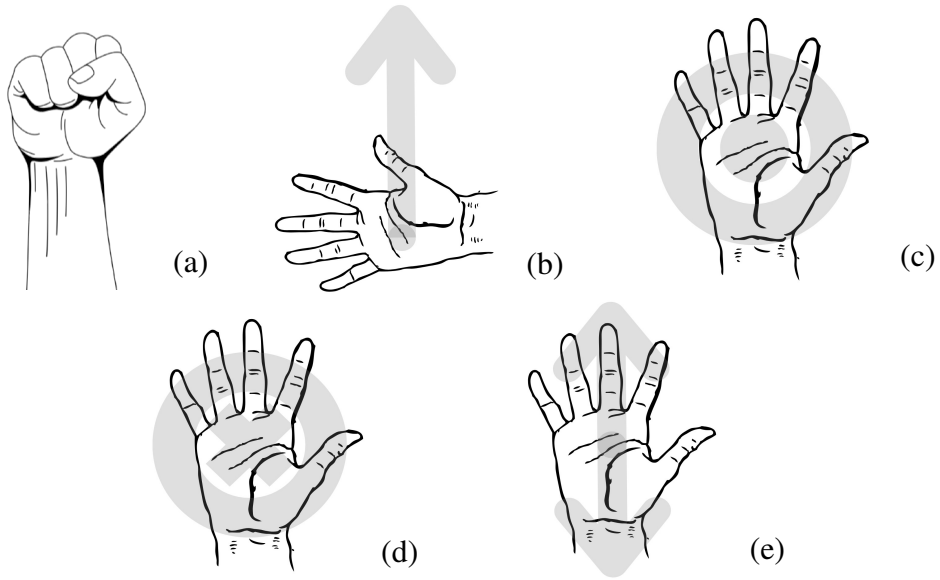


Figure 4.2: The five types of hand gestures used in the implementation

4.2.2 Number of Participants

The number of participants involved in a HGR system significantly influences the robustness of the classification algorithm. More participants lead to different hand characteristics, which mitigates the risk of the machine learning algorithm overfitting to a single participant. Furthermore, it allows for the collection of a larger and more diverse dataset, enabling the model to generalise more effectively. Despite these facts, due to ethical considerations and potential risks associated with machine learning algorithms, the utilisation of multiple participants was unfeasible for this particular implementation. Consequently, the proposed hand-gesture dataset was generated by a sole participant, the author of this project.

4.2.3 Dataset

Each gesture was repeated 150 times, leading to a collection of 750 samples in total. While this quantity may appear relatively small when compared to the larger data samples found in the current state-of-the-art, this decision was deliberate. This choice was made due to the considerable time and effort involved in recording and processing the data before feeding it into the machine learning algorithm, particularly when the experiment only had one participant.

To address the limitation of the dataset's size, data augmentation techniques were employed. These techniques served to expand the dataset, enhancing the robustness of the machine learning classification algorithm. Further details on these methods are discussed in Section 4.7.3.

4.3 Experimental Hardware

4.3.1 AWR1843 with DCA1000EVM

The radar chip employed in this implementation is the AWR1843 model by Texas Instruments. According to the datasheet, this device is described as an 'integrated single-chip FMCW radar sensor' capable of functioning within the 76-81 GHz frequency range. Key characteristics of the AWR1843 are provided in Table 4.1 below.

AWR1843	
Frequency Range	76-81 GHz
Number of Receivers	4
Number of Transmitters	3
Maximum ADC Sampling rate	25 Msps
Available Bandwidth	4 GHz

Table 4.1: Key characteristics of the AWR1843 radar chip

The DCA1000EVM, another Texas Instruments product, serves as the capture card used to interface with the AWR1843. This device facilitates the streaming of ADC data over Ethernet. In this

implementation, the AWR1843 radar chip collaborates with the DCA1000EVM data capture card to gather raw radar data.

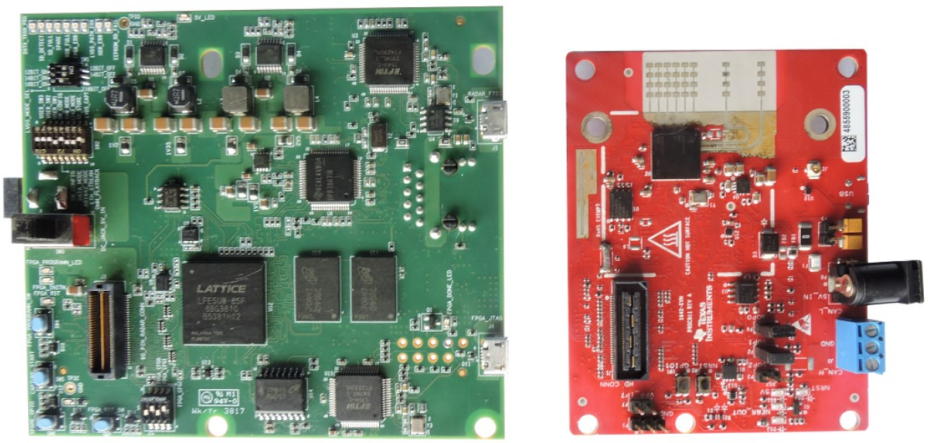


Figure 4.3: DCA1000EVM and AWR1843 Boards, respectively

4.3.2 Intel i7 NUC

The radar module, consisting of the AWR1843 and the DCA1000EVM, was interfaced with a Intel i7 Next Unit of Computing (NUC). This device is a compact yet robust computer that can run the Linux operating system, facilitating user interaction with the radar module. Through this computer, the setup and configuration of the radar system can be performed. Additionally, it enables the storage and conversion of recorded raw radar data, discussed further in Section 4.5.3.

4.3.3 Additional Hardware

The experimental hardware setup also includes a laptop dedicated to processing the system. While the Intel i7 NUC is capable of processing, a laptop was chosen due to its greater versatility and functionality compared to the NUC. The laptop is employed for receiving the raw radar data, which is wirelessly transferred from the NUC, with details regarding this transfer method provided in Section 4.5.3. Additionally, this laptop is utilised for the desgning, training and testing of the hand gesture classification model. The specific laptop used for is an Acer Nitro 5 Intel Core i5-11400H.

4.3.4 Other Options and Selection Process

There are various options available for the radar system, as discussed in Section 2.2.1, each of which has a distinct impact on radar performance. Among these options, the AWR1642, IWR1443 and IWR6843 stand out as notable choices due to their multiple receivers, compatibility with the DCA1000EVM and wide bandwidth capabilities.

The choice to employ the AWR1843 and DCM1000EVM in conjunction with the Intel i7 NUC was primarily influenced by the availability of these components in the lab, nullifying the need for any additional purchases. This practicality factor played a significant role in the selection. Especially due to the fact that these components are out of this project's budget. To provide context, as of October

2023, the AWR1843 costs USD 300 (or ZAR 5750) [87], while the DCA1000EVM is priced at USD 600 (or ZAR 11500) [88].

Section 3.2.1, particularly in equation 3.7, highlights the importance of a higher available bandwidth for achieving improved range resolution. In the context of HGR, it is crucial to maximise range resolution, as it enhance the radar's ability to capture subtle variations in the range of targets. The AWR1843, with its 4GHz bandwidth, is a radar chip that has been widely used in previous HGR implementations. Additionally, its inclusion of four receiver antennas has proved to be beneficial to the radar's performance.

4.3.5 Overall System

An illustration of the overall radar system is displayed in Figure 4.4.

Figure 4.4: Picture of the radar system utilised for the project

4.4 Software Selection

The processing pipeline was developed using MATLAB, a high-level programming language known for its rich array of built-in functions. MATLAB is a widely adopted choice in the field of radar and signal processing due to its comprehensive plotting toolboxes and capabilities for data manipulation and visualisation. Following the conversion process described in Section 4.5.3, the raw radar data is stored in a Hierarchical Data Format version 5 (HDF5) file. The MATLAB library provides a range of functions for reading and extracting data from HDF5 files, making it adequate for this implementation. Additionally, MATLAB offers mathematical functions that are useful for the operations conducted in the processing of raw radar data.

In conjunction with MATLAB, Python played an vital role throughout the system. Python's simplicity and versatility facilitated the development of utility scripts which allowed for easier transitions between different phases of the design. Additionally, PyTorch was utilised for the machine learning stages of the design. PyTorch, an open-source machine learning framework, facilitates the development and training of machine learning algorithms, particularly neural networks. The choice of PyTorch over alternatives such as Keras or TensorFlow was primarily from personal preferences and prior experience.

4.5 Data Collection

4.5.1 Radar Configuration

The participant was seated in a chair and executed hand gestures within a range of 20.0 cm to 30.0 cm in front of the radar, which was positioned on a desk in the laboratory. The radar was oriented towards a white wall and other desks, which had the potential to be misinterpreted as targets and generate signal reflections that could impact radar data. However, an attempt to mitigate these challenges was done during data processing through the application of clutter removal methods, which will be elaborated in Section 4.6.6.

On the software configuration side, the Intel i7 NUC played a pivotal role by providing an interface for radar interaction. This allowed for the initialisation of radar parameters essential for the experiment.

4.5.2 Parameter Selection

The selection of parameters was a pivotal aspect in the data collection phase of this design. Several radar parameters were of importance, including:

- Starting frequency
- Total Bandwidth
- Transmit/Receive antennas
- Bandwidth used
- Idle time
- Frequency Slope
- Sampling rate
- Number of ADC samples per chirp
- Number of chirps per frame
- Number of frames
- Chirp duration (frame periodicity)

These parameter choices were guided by several factors, such as the nature of the target (in this case, hand gestures), the environment and duration for the gestures to be performed.

Some parameters were selected to align with previous literature, as they yielded optimal performance. Conversely, others were determined through calculations and extensive testing to arrive at the most appropriate values. These calculations used radar theory equations, detailed in Section 3.2, and allowed for the determination of the required maximum range, maximum velocity, and resolutions. A goal for this implementation was to maintain consistency between the parameters used for the different hand gestures. Therefore, these parameters remained consistent across all samples in the dataset. The results of these calculations are presented in Table 4.2 below.

Calculated Parameters	
Parameter	Value
Maximum Unambiguous Range	4.9 m
Maximum Unambiguous Velocity	0.04 m
Range Resolution	3.9
Velocity Resolution	0.24

Table 4.2: Performance parameters of the radar

Furthermore, the number of antennas employed, the starting frequency, and bandwidth utilisation were gathered using radar theory principles discussed in Section 3.2. To obtain the remaining parameters, the Texas Instruments document regarding chirp parameter programming [65] was an invaluable resource. This document facilitated the determination of parameters such as the number of ADC samples per chirp, the quantity of chirps, frames, sampling rate, idle time, chirp duration, and ramp time.

Before data collection, all these parameters were tested using Texas Instruments' online tool, the mmWave Demo Visualiser [89]. This comprehensive process yielded the parameters that were ultimately utilised in configuring the radar setup. Table 4.3 below illustrates all the parameters utilised for the setting up of the radar system. Subsequently, the radar was prepared for the acquisition of raw data.

Radar Parameters	
Parameter	**Value**
Starting Frequency	77 GHz
Bandwidth	4 GHz
Channels	3Rx - 4Tx
Used (RF) Bandwidth	3.44 GHz
Idle Time	186 μs
Ramp Time	57.14 μs
Frequency Slope	70 GHz/s
Sample Rate	2279
Number of ADC samples	112
Number of Chirps	32
Number of Frames	50
Frame Period	70 ms

Table 4.3: Radar parameters utilised for the implementation

4.5.3 Data Conversion and Transferring

Upon capturing hand gestures, the raw radar data was saved in the form of a single binary '.bin' file on the Intel i7 NUC. The structure of the binary file for an AWR1843 using the DCA1000EVM can be

illustrated through the diagram presented in Figure 4.5, provided by Texas Instruments [90].

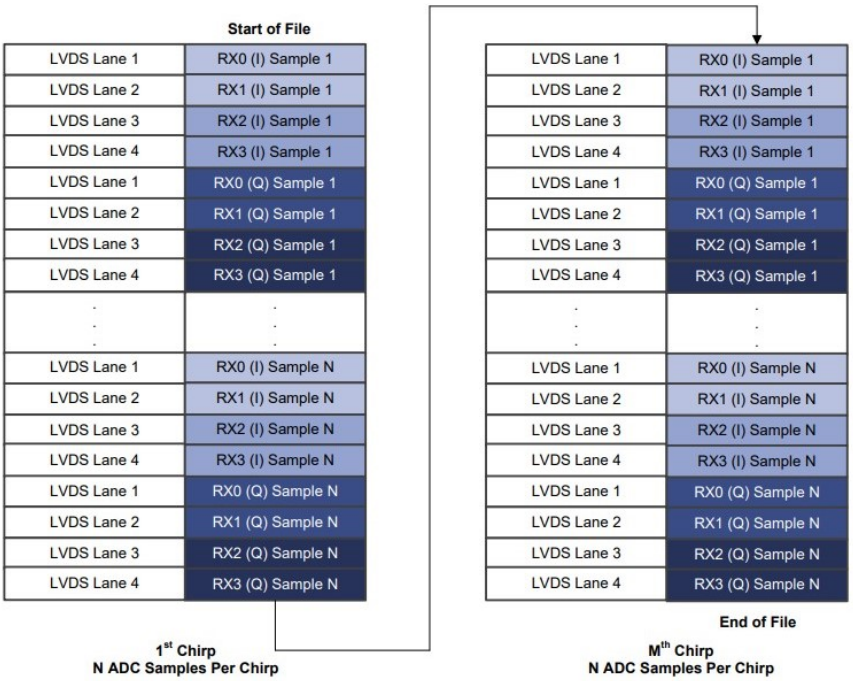


Figure 4.5: Complex data format for AWR1843 using DCA1000EVM [90]

The data was subsequently converted into the HDF5 file using a Python script available on the Intel i7 NUC. This Python script, previously developed by Nicholas Bowden in a prior experiment [91], transforms the data and generates a new file as illustrated in Figure 4.6. The HDF5 format is specifically designed to accommodate large and complex heterogeneous datasets, making it highly suitable for radar data.

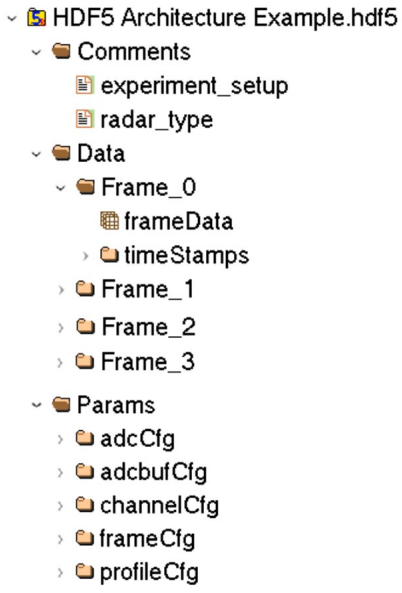


Figure 4.6: Complex data format for HDF5 file

Following the conversion process, the data is transferred to the laptop for signal processing. This transfer is accomplished using 'SFTP,' a tool that enables synchronisation between the local directory of the laptop and a remote directory (Intel i7 NUC). This tool facilitates wireless data transfer between the Intel i7 NUC and the laptop, facilitating data retrieval on the laptop for subsequent signal processing and machine learning phases of the design.

4.6 Processing pipeline

4.6.1 Pipeline Overview

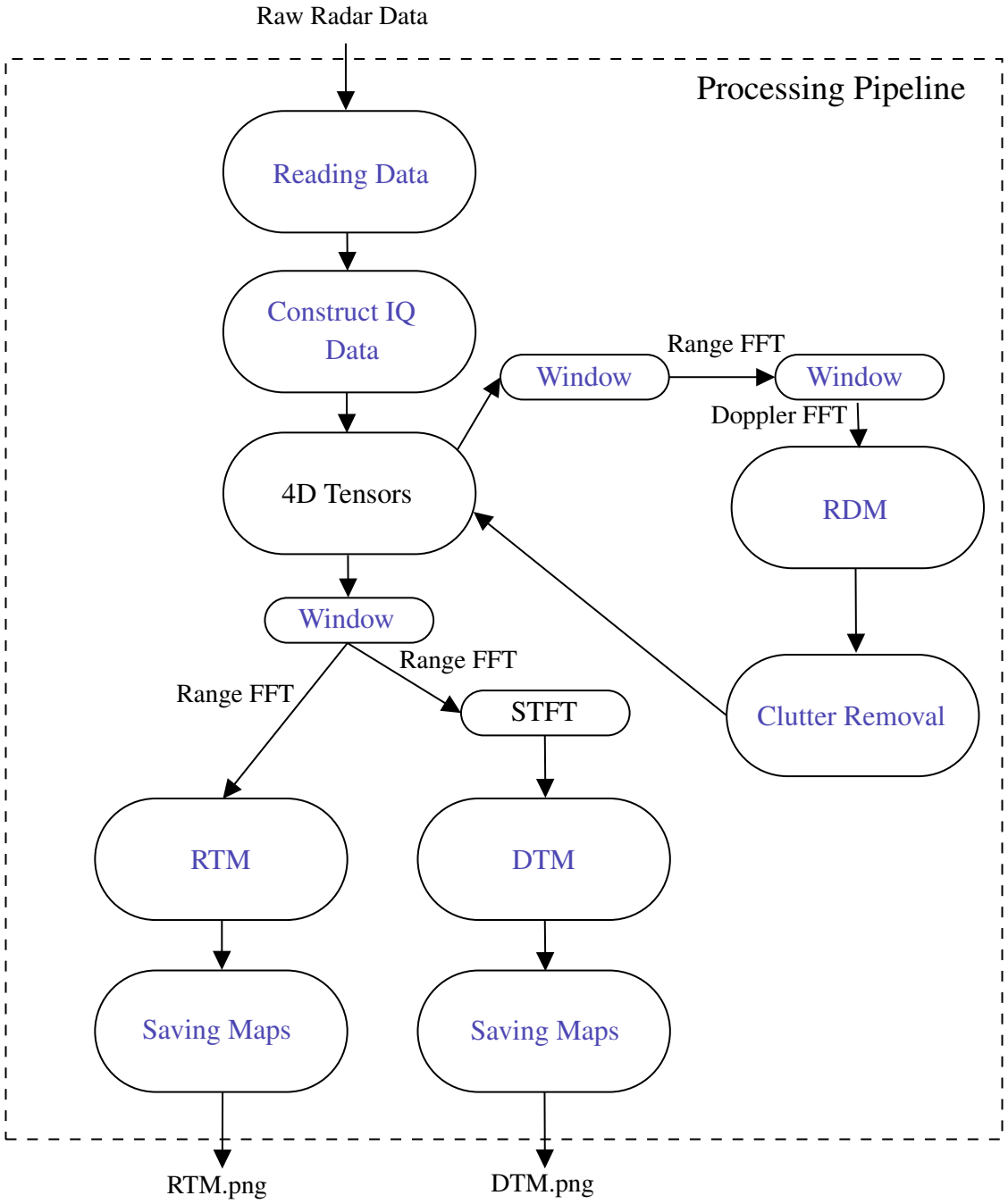


Figure 4.7: Block Diagram of the entire processing pipeline

4.6.2 Reading the data from HDF5 file

Once the raw radar data was acquired on the laptop, the relevant radar parameters were extracted by reading data from the HDF5 file in MATLAB. This operation utilised the 'h5read' function, requiring two parameters: the filename of the HDF5 file where the data is stored and the name of the dataset to be retrieved. Through this function, the radar information was acquired such as the number of frames, chirps, samples, channels, sampling rate, sampled bandwidth, and chirp duration.

4.6.3 Construction data cube

From the raw radar data, the relevant radar information was gathered for each frame to construct a 3D data tensor, as discussed in Section 3.3. To extract this 3D tensor data from each frame, the 'h5read' function was employed again. However, the data obtained from the HDF5 file is initially in the form of a binary data array. Therefore, it was transformed into complex data by separating the different indices. This required reassembling the data to create complex information, a process accomplished through a dedicated function designed to construct In-Phase Quadrature (IQ) data from raw radar data. This is elaborated upon the example in Figure 4.8 below.

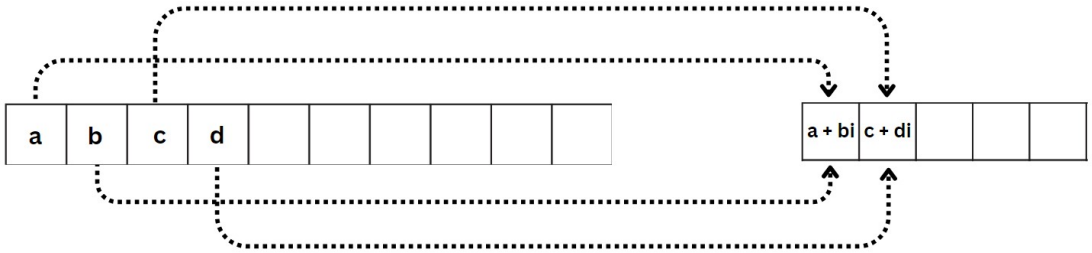


Figure 4.8: Diagram showing how the IQ data is constructed from raw radar vector

Subsequently, time frame profiles were constructed, involving the placement of the complex data into a multidimensional matrix structure. This structure is the 3D tensor previously discussed in Section 3.2, which contains information on channels, chirps, and samples for each frame. All these 3D tensors were then organised into a 4D tensor by adding the 'frames' dimension to create the format [channels x frames x chirps x samples], containing the entirety of the required data. Interacting with a single tensor with all the information simplified data manipulation and facilitated effective processing and map generation.

4.6.4 Processing Maps selection

Numerous processing techniques and derived maps can be generated from raw radar data. These maps are essential for visualising the data in graphical form and serving as input for the machine learning algorithm. Therefore, it is important to determine the appropriate number of maps, their optimal format, and the information they convey before designing them.

It was concluded that the chosen maps should be over a specific time frame, given that hand gestures

occur over a period rather than instantaneously. This aspect enables the analysis of changes over time, significantly enhancing the maps' ability to provide relevant features for the machine learning algorithm. The selected maps for processing were the RTM and DTM. As elaborated in Section 3.3, these maps offer valuable insights into changes in target range and velocity over time. This choice was reinforced by the common use of these two maps in prior research.

It was also discussed in radar theory that the DTM plotting involved selecting range bins corresponding to the hand gesture's operating range. To identify the appropriate range bin, the RDM was generated. The RDM serves the additional purpose of visualising clutter and noise in the data, which is crucial for implementing clutter removal techniques. Minimising clutter in the data results in a clearer representation, making the maps more distinct and improving the classification algorithm's ability to identify features in different types of gestures.

4.6.5 Range-Doppler Maps

After constructing the data cube and generating time frame profiles, a 4D data tensor was obtained. This tensor was further manipulated to create a RDM. As detailed in the theory Section 3.3.3, the generation of these maps can be accomplished by either selecting a single channel or by performing a complex summation of data across all channels. In this implementation, both methods were evaluated to determine which one produced clearer maps.

It was found that the summation of all channels was the more effective approach, primarily because it reduces the Signal to Noise Ratio (SNR) of the data. This reduction enhanced the visibility of the target and made it easier to distinguish from noise. The difference between selecting a single channel and using the summation of all channels is visually demonstrated in Figure 4.9, where the target at 0.8 meters is notably easier to discern and identify.

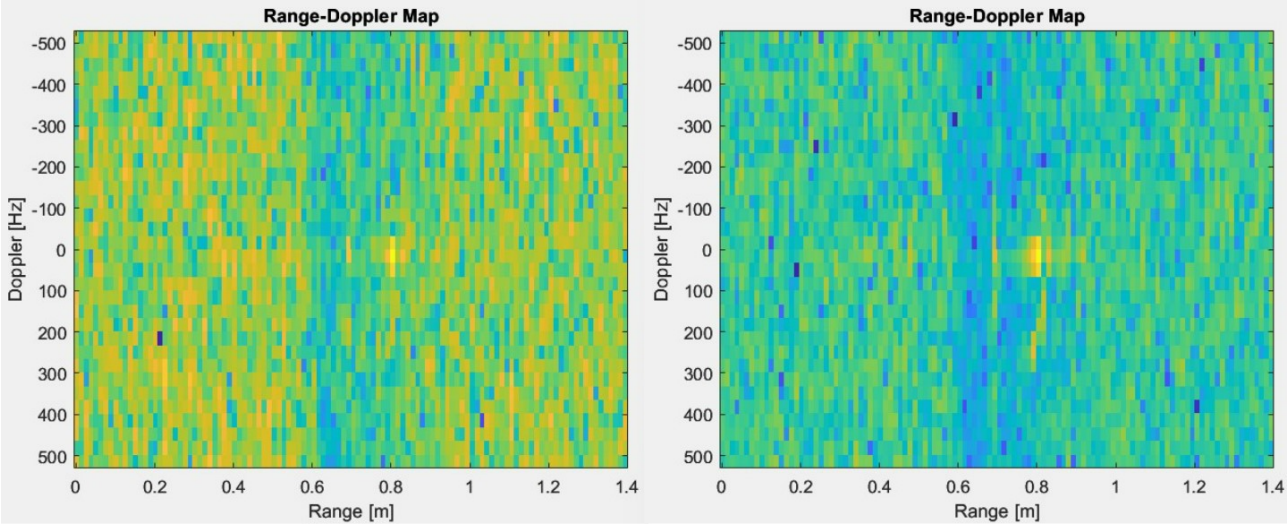


Figure 4.9: Difference of choosing one channel or summing the channels together

Subsequently, a single frame was selected to create a data matrix consisting solely of chirps and samples. The same process as described in the theory Section 3.3.3, was followed. The range FFT was executed by applying aFFT to the fast-time domain (samples) to acquire range information. Following that, a

FFT was performed by employing a slow-time FFT on the slow-time domain (chirps) to extract velocity information. The resulting 2D matrix, containing data for range in meters and velocity (Doppler) in Hertz, was then visualised by calculating the absolute value of the data.

4.6.6 Clutter Removal

Having identified the clutter within the data and the segments that could be eliminated using the RDM, a clutter removal method was designed. This method involved performing complex subtraction on specific data segments.

To initiate the clutter removal process, a 3D tensor was constructed for the first 10 frames of data, wherein no hand gestures had been performed yet. These initial 10 frames encompassed the consistent clutter and noise inherent in the data. To eliminate this unwanted data, the 3D tensor for the first ten frames was initially averaged, and then this average was subtracted from the subsequent frames, ranging from frame 11 to 'n,' where 'n' denotes the total number of frames. This process is graphically depicted in Figure 4.10.

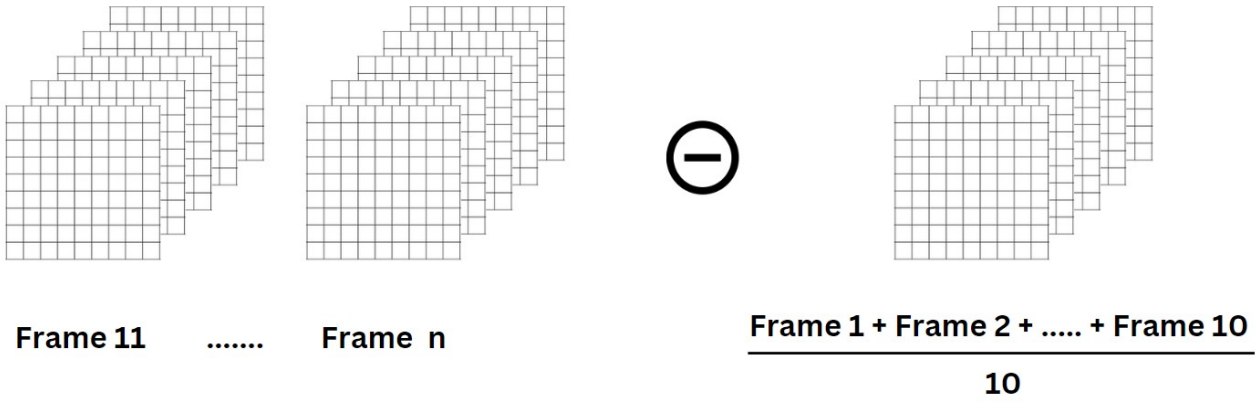


Figure 4.10: Clutter removal process

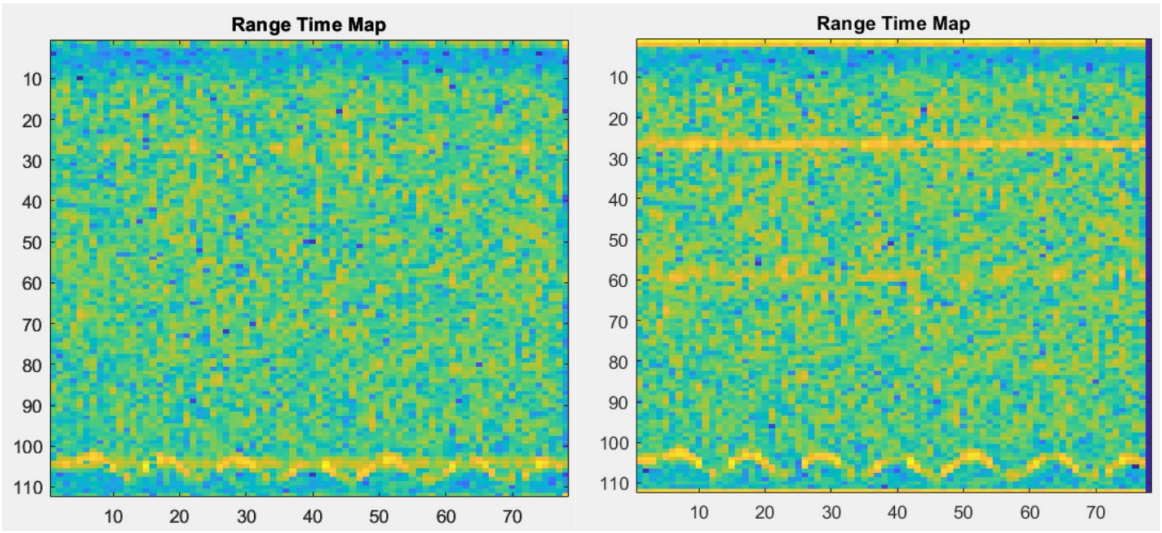
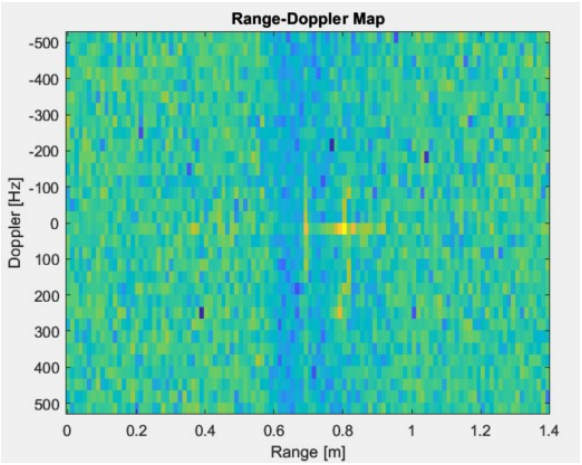


Figure 4.11: Example of RTM with and without clutter removal

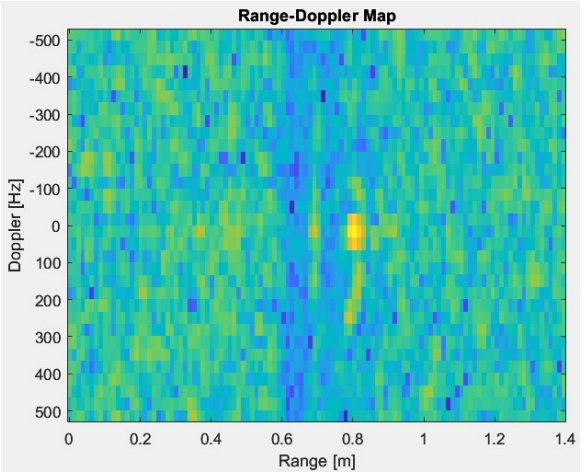
By applying this technique, undesired signals were effectively removed, thereby enhancing the distinction and detection of hand gestures. It also mitigated false radar signal detections, enabling the accurate selection of range bins. The advantages of clutter removal in the signal are evident in Figure 4.11, which displays a RTM with and without the clutter removal process implemented. In this figure, the clutter, which is depicted as a horizontal line representing a stationary target at a specific range over time, is noticeably absent. This clutter could represent objects such as walls or other targets that the radar was initially detecting.

4.6.7 Windowing

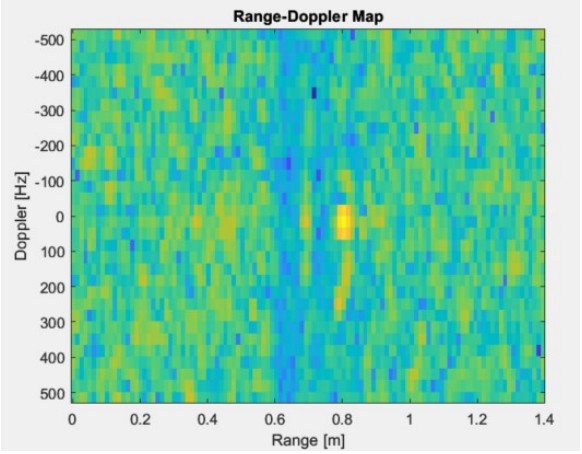
The windowing process was designed to be applied to the entire dataset prior to any FFT operation. The first windowing step was conducted before the Range FFT. After obtaining the 3D tensor from the raw radar data, each sample from every chirp and channel was multiplied by a specific window function. Several types of windowing functions were tested, including 'Hamming' and 'Blackman' windows. A comparison of the various windowing functions and their effects on the data is presented in Figure 4.12.



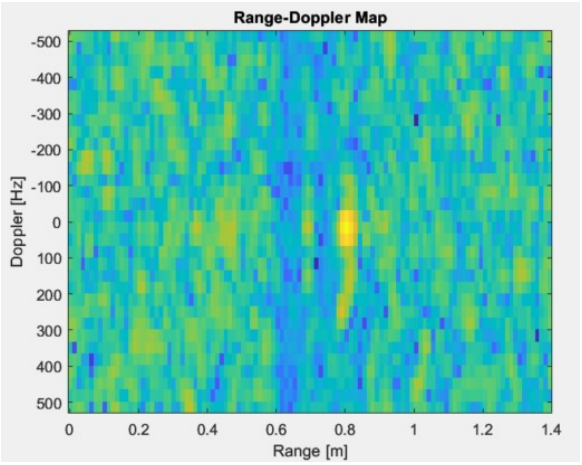
(a) RDM using no windowing



(b) RDM using Hann windowing



(c) RDM using Hamming windowing



(d) RDM using Blackman windowing

Figure 4.12: RDM with different windowing functions

The chosen window function for the Range FFT was the 'Hann' window function. To maintain consistency throughout the processing, it was decided to use the same windowing function for all subsequent FFT operations. The selection of the 'Hann' window function was based on its advantageous attributes, such as improved frequency resolution and reduced spectral leakage. This method of testing and selecting the window function was also applied to the Doppler FFT, where the 'Hann' function was chosen once again.

4.6.8 Range-Time Maps

The construction of the RTM was performed according to the principles of radar theory outlined in Section 3.3.2. Following the channel summation, the data took the form of a 3D tensor, encompassing frames, chirps, and samples. Initially, a FFT was computed over the fast-time domain to obtain target range information, as previously discussed. Prior to this FFT, windowing was applied to the data. The data can be analysed either on a per-chirp basis or by considering the summation of all chirps. After thorough testing, it was observed that there was minimal difference between selecting a specific chirp and summing all chirps. Choosing a single chirp and reducing the data to a 2D matrix, which includes samples for each frame, allowed for the computation of the absolute value of the data, resulting in the generation of the RTM.

Since the data of interest only occurs within a few specific range bins, the maps were be cropped to eliminate the extra information. This is achieved by visualising only the desired range over time.

4.6.9 Doppler-Time Maps

The DTM was constructed in accordance with the principles described in Section 3.3.4. The range bins of interest were identified through the RDM. They are a prerequisite for the computation and visualisation of the DTM. Given that the Range FFT had already been applied for the visualisation of the RTM, the results of that operation could be repurposed for the DTM.

The individual samples within the 3D tensor represented the range bins, and these values underwent a complex summation to further reduce the tensor's dimension. This process led to a matrix, which included frames and chirps corresponding to the specific range bins. Subsequently, this matrix was further simplified into a single vector by concatenating the data from each chirp into a vector for each frame. This resulted in a vector with dimensions of [1 x (frames * chirps)]. This vector played a crucial role in facilitating the use of a sliding window, as discussed in Section 3.3.4.

Following the process outlined in Figure 3.12, the DTM was successfully generated. However, essential parameters such as the window size and overlap needed to be fine-tuned through multiple tests to find which configuration yielded the best results. The sliding window parameters remained consistent across various hand gestures. Lastly, a padding factor was introduced after the STFT process. This padding factor served to enhance frequency resolution and contribute to smoother DTMs. These parameters are illustrated in Table 4.4 below.

Spectrogram parameters	
Parameter	**Value**
Window Size	30
Overlap	99%
Padding Factor	10

Table 4.4: Parameters used for the DTM

4.6.10 Saving the Maps

After generating the two maps, they were stored directly from Matlab into a designated folder on the laptop. The maps were saved sequentially using Matlab's 'saveas()' function. Specifically, the initial sample data for hand gestures, consisting of the RTM and DTM , were saved as 'rtm_i.png' and 'dtm_i.png,' with 'i' representing the sample number corresponding to that particular type of hand gesture. This process was repeated for all the samples within each of the five distinct hand gesture categories.

4.7 Machine Learning Process

4.7.1 Classification Algorithm Selection

Given the large array of machine learning algorithms employed in prior HGR research, it is vital to conduct comprehensive comparisons and assessments to determine the most suitable algorithm. Previous studies have utilised various algorithms, resulting in distinct outcomes and performances. The classification algorithm selection process was done in parallel with the development of the processing pipeline. This approach was necessary as knowing which algorithm would be employed was crucial for designing the appropriate input. The selection process started with the evaluation and analysis of system requirements. Subsequently, various options were identified, and their respective advantages and disadvantages were assessed to select the most appropriate algorithm.

Classification Algorithm requirements

The machine learning algorithm utilised in this project is tasked with performing a classification task, specifically aimed at accurately categorising five distinct hand gestures. In order to meet project objectives, this algorithm must meet certain requirements. These essential requirements for the classification algorithm are as follows:

1. Successfully classify a minimum of five hand gesture types with an accuracy rate greater than 90% on unseen data.
2. Accept inputs in the form of images.
3. Possess the ability to facilitate the training and implementation of new gesture categories.

Feature selection

When considering feature selection and data input for the machine learning algorithm, various options were explored. As outlined in the literature review in Section 2.2.3, one approach involves utilising a single map as the feature, while another strategy incorporates multiple maps as distinct inputs in a multi-channel classification algorithm. However, for this project, a different approach was adopted: a multi-dimensional input that integrates dimensions from multiple maps into a single image, which is then fed into a single-channel algorithm. This method was inspired by Zhao et al.'s innovative approach in 2022 [49], where multiple maps were successfully combined into a single image. Following Zhao et al.'s approach, this project aimed to merge maps generated by the processing pipeline into a single combined image, which would serve as the input for the classification algorithm. This merging process involved combining the RTM and the DTM to create a single image.

Classification algorithm Options

As discussed in Section 2.3 of the literature review, numerous machine learning algorithms have been applied to the HGR classification task. Among these, three classification algorithms have consistently yielded high performance while satisfying all the previously mentioned requirements. The three algorithms investigated were CNN, LSTM, and SVM. Although any of these algorithms could be suitable for this implementation, a CNN was chosen. This selection was primarily driven by the fact that CNNs excel at recognising and capturing spatial and velocity features within radar maps, a crucial aspect in HGR. Additionally, CNNs facilitate automatic feature removal, eliminating the need to design feature removal techniques within the model, a requirement often associated with SVM algorithms. Moreover, CNNs handle large datasets, allowing for improved generalisation. While LSTMs are usually more effective for identifying features with a time dimension, the dominant features in the chosen maps are primarily the range and the velocity of the hand-gestures. Furthermore, CNNs are well-established, relatively straightforward to implement while still achieving high accuracy performances. Lastly, the choice of a CNN was guided by both preferences and personal past experiences with this type of model.

4.7.2 Combining the maps

Upon the decision to merge the maps and feed them into the CNN model, a method for their combination was designed. This process was executed using a Python script specifically developed for this task. Given that the images generated from the Matlab document shared identical pixel dimensions, it was a straightforward task to uniformly crop them within the code. Subsequently, the cropped images were assembled into a single image and saved. This procedure was iterated for each set of sample data corresponding to various types of hand gestures, resulting in a total of 750 images. These images were employed as the input data for the machine learning algorithm. An example of this combined image is presented in Figure 4.13 below.

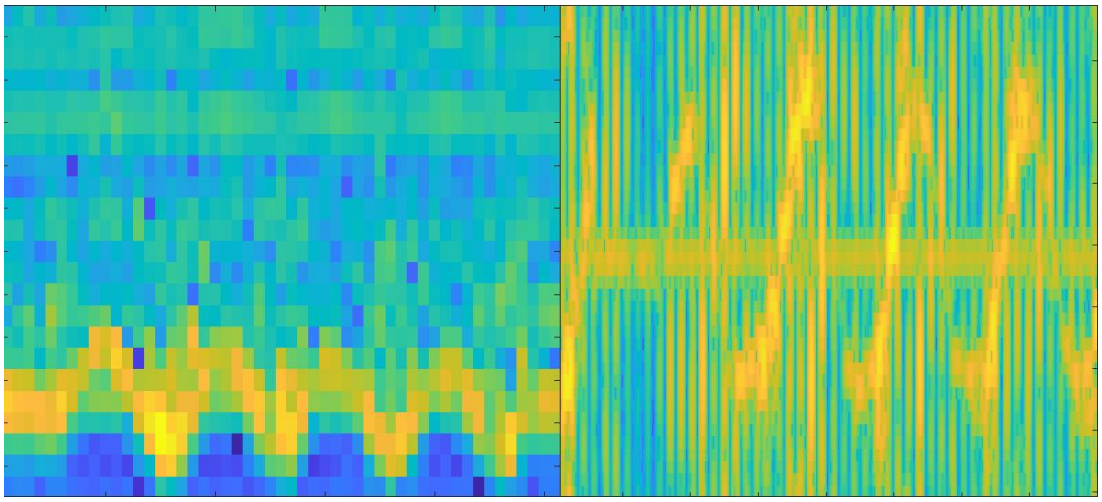


Figure 4.13: Example of a combined image of a RTM and DTM

4.7.3 Data Manipulation

Before designing the CNN algorithm, the dataset was divided into three distinct sets. The data was split in a 60:20:20 ratio, with 60% allocated for training the model, 20% for validation, and another 20% for testing. This distribution means that out of the 750 collected and processed hand gesture samples, 450 were used for training, 150 for validation, and 150 for testing.

Due to the dataset being exclusively collected from a single participant, the author, its size was limited. To enhance the CNN model's ability to generalise effectively, a concept discussed in Section 3.4.4, data augmentation techniques were employed to expand the dataset within the CNN model. This process included rotating the images by 90°, 180° and 270°, as well as resizing them to different scales. This augmentation improved the model's robustness, enabling it to perform effectively and achieve high accuracy even when presented with new data.

4.7.4 Machine learning algorithm design

Model Architecture

The CNN model was designed within the PyTorch framework, which provides a rich set of tools for developing machine learning models. The model's initialisation was based off the CNN theory, discussed in a previously in Section 3.4. The model's architecture was established using PyTorch's 'nn.module' base class, initialising it with three convolutional layers followed by three batch normalisation layers. These layers were initialised using the 'nn.Conv2d' and 'nn.BatchNorm2d' functions offered by the Pytorch base class, respectively. Their purpose was to identify and extract the features from the input maps, enabling the distinction of different aspects in each gesture.

To reduce the dimensions of the feature maps, a max-pooling layer was applied after each convolutional layer using the 'nn.MaxPool2d' function. Following the final convolutional layer, the feature maps were flattened into a one-dimensional vector using the 'nn.Flatten' function. Subsequently, the hidden layers of the system were introduced for further processing. This was achieved using the 'nn.Linear'

function, connecting the flattened layers to fully connected layers. Each hidden layer was subsequently passed through another batch normalisation layer to ensure proper array normalisation.

The parameter configuration within each function was determined using a convolutional network shape calculator tool [92]. This tool aided in establishing the output sizes after each layer, ensuring the selection of appropriate inputs. Notably, the size of the input images was 1187 x 534 prior the resizing of the augmentation technique.

The forward pass of the model specified how the input maps traverse each layer and in what order. This process was facilitated using ReLU activations. The final layer of the model, the third and ultimate fully connected layer, was responsible for generating the model's output.

The CNN model architecture is visually represented in Figure 4.14.

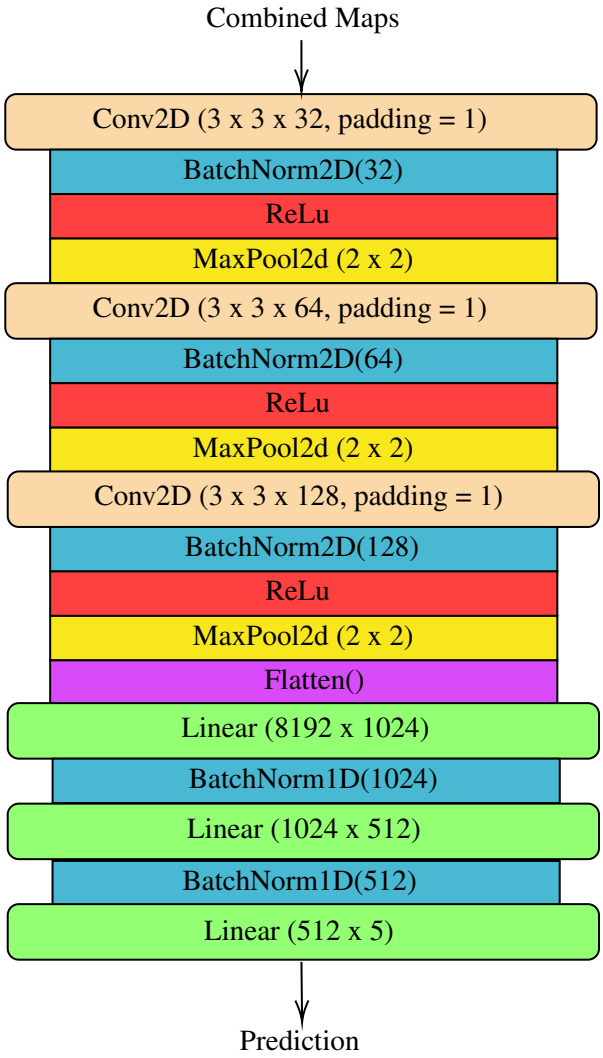


Figure 4.14: Proposed CNN Model Architecture

Model Training

The training of the CNN model involved several steps within the context of the backpropagation algorithm. A loss function, specifically the cross-entropy loss, and an Adam optimiser were defined in the model. The training dataset, comprising images and their corresponding labels, was loaded to start the training of the model. Initially, training parameters' weights were randomly initialised. The input images were subsequently subjected to convolution with randomly weighted matrices and propagated through the model's layers, resulting in random output predictions. Based on the predicted output and the actual output, the cross-entropy loss function was computed. As discussed in Figure 3.17, this loss value was then utilised to update the weights through the backpropagation process using the Adam Optimiser. This process was repeated for different hyperparameters in order to obtain the best possible performance. After the tuning of the hyperparameters, the best-performing model on the validation set was selected for further testing on an independent testing set. This step was essential to confirm the model's robustness and generalisation capability while preventing overfitting.

The selected final hyperparameters for the model are presented in the Table 4.5 below.

Hyperparameters	Value
Learning Rate	0.0001
Learning Rate Decay	0.1
Learning Rate Decay Step	7
Weight Decay	1E-06
Batch Size	64
Number of Epochs	10

Table 4.5: Hyperparameters used for the proposed CNN model

4.8 Chapter 4 Summary

This chapter initiated the structure for the entire design process, highlighting the various decisions made throughout the project and the specific approach employed for each subsystem. It commences by providing an overview of the complete design process, breaking down the system into iterative tasks. The chapter delves into the design of various types of hand gestures, encompassing the five distinct categories: grabbing, lifting, pulling, pushing, and patting. In the context of constructing the dataset, it was decided to utilise a single participant to collect a total of 750 samples. The selection of hardware components was then discussed, with the choice of the AWR1843 radar with the DCA1000EVM, and an Intel i7 NUC. The chapter also examines the radar configurations and the parameters selected to set up the radar system. Following this, the processing pipeline was explored, detailing the creation of three essential maps: RDM, RTM and DTM. Techniques such as windowing and clutter removal were employed within the processing pipeline to enhance the quality of information in each map. The choice of combining the maps into a single image was then discussed. Ultimately, the chapter concluded with the design of the CNN model and a discussion of the model training process.

Chapter 5

Results and Discussions

This chapter of the report focuses on the project's results. The experiments and testing of the implementation were divided into two main sections. The first part evaluates the outcomes of the processing pipeline, which involves analysing different sample combined images for each of the five hand gestures. It begins by discussing the expected outputs of the RTM and DTM, which were combined into a single image, for each particular hand gestures. The obtained maps are subsequently evaluated and analysed.

The second part of the results section delves into the performance of the proposed classification algorithm. Initially, the loss function is assessed alongside the confusion matrix. Subsequently, an assessment is made between the performance of the proposed model and existing systems.

5.1 Processing Pipeline Results

This section of the results provides an in-depth analysis of each hand gesture type and the combined images generated for them, which will be utilised as inputs for the CNN algorithm. Each map offers unique characteristics that can be distinguished by examining the variations in range and Doppler over time. The combined images represent a fusion of the RTM, on the left and the DTM, on the right, derived from the radar data of the hand gestures.

Before analysing the various images, it's crucial to note that, for the RTM on the left side, the origin is on the bottom left corner of the plot. The radar is positioned at this origin, meaning that the higher a target appears on the map, the farther it is from the radar.

5.1.1 Grabbing Hand-gesture

In the context of a grabbing hand gesture, minimal variations in range can be anticipated. This is due to the fact that only the fingers of the hand move when closing the fist to perform the gesture and the range resolution is not low enough for the radar to capture this change. During the closing of the fist, there should be a change in the velocity of the target. This change in velocity should occur over a short period of time, as the gesture is quick and does not span across all the frames. Figure 5.1 below illustrates a sample combined image obtained representing the grabbing hand gesture.

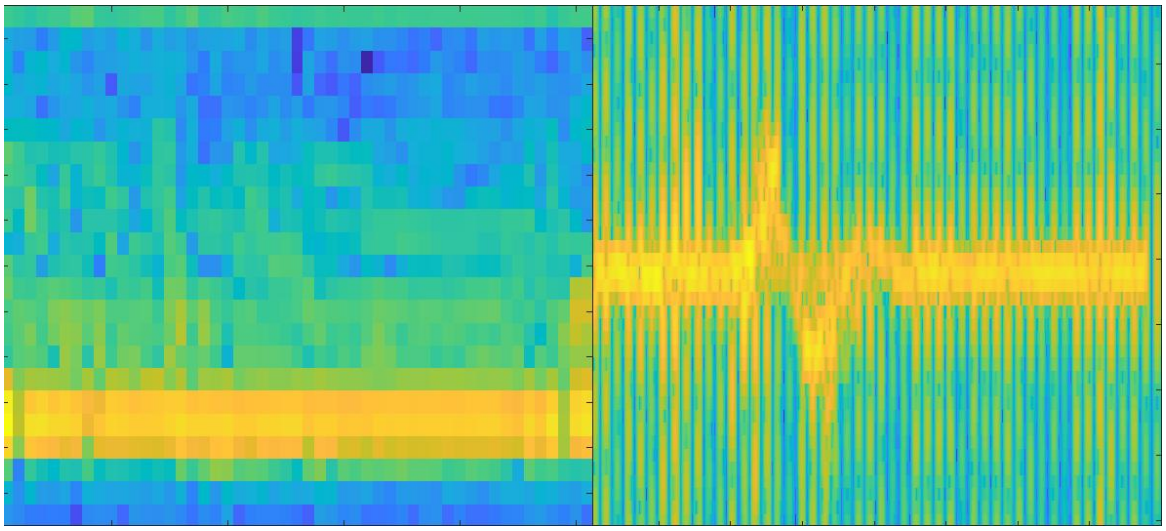


Figure 5.1: Combined image example of the grabbing hand-gesture

Figure 5.1 aligns with the expectations for a grabbing hand gesture. It indeed reveals minimal changes in the range of the target. The combined image also displays a noticeable alteration in velocity before stabilising again, which is in line with the characteristics expected from a grabbing gesture.

5.1.2 Lifting Hand-gesture

Concerning lifting hand gestures, several characteristics can be anticipated. Firstly, with respect to the range of the target, it is expected that the target's range will remain constant, as the target is not moving closer to or farther away from the radar. Additionally, a small change in velocity should be observable in the DTM. Figure 5.2 below illustrates a sample combined image obtained representing the lifting hand gesture.

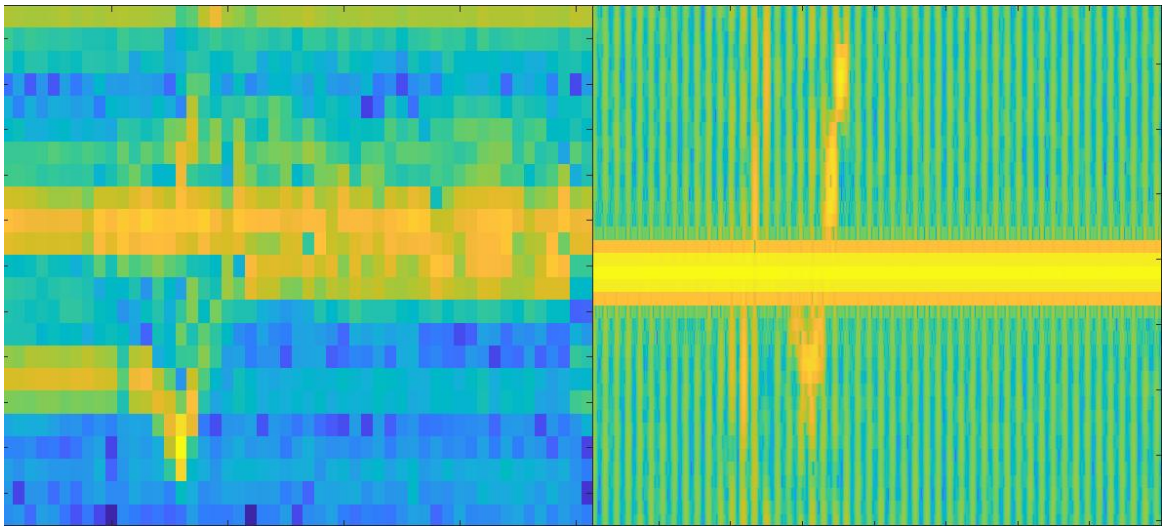


Figure 5.2: Combined image example of the lifting hand-gesture

Figure 5.2 appears to somewhat align with the expected outcomes from radar data of a lifting hand gesture. The target's range remaining constant can be observed; however, it seems that the radar also detects a second target during the gesture. This is caused by the radar identifying the participant's arm as a distinct target. In terms of velocity changes, Figure 5.2 accurately represents what can be expected from a lifting hand gesture.

5.1.3 Pulling Hand-gesture

In the context of the pulling hand gesture, certain expectations can be outlined. Firstly, regarding the range of the target, it is expected that the range will undergo a change and then remain constant, with the target appearing to have moved away (up on the map) from the radar. In terms of the target's velocity, the map should exhibit a change in velocity while the hand is in motion from one range to another, followed by a period of stability with constant velocity. Figure 5.3 below illustrates a sample combined image obtained representing the pulling hand gesture.

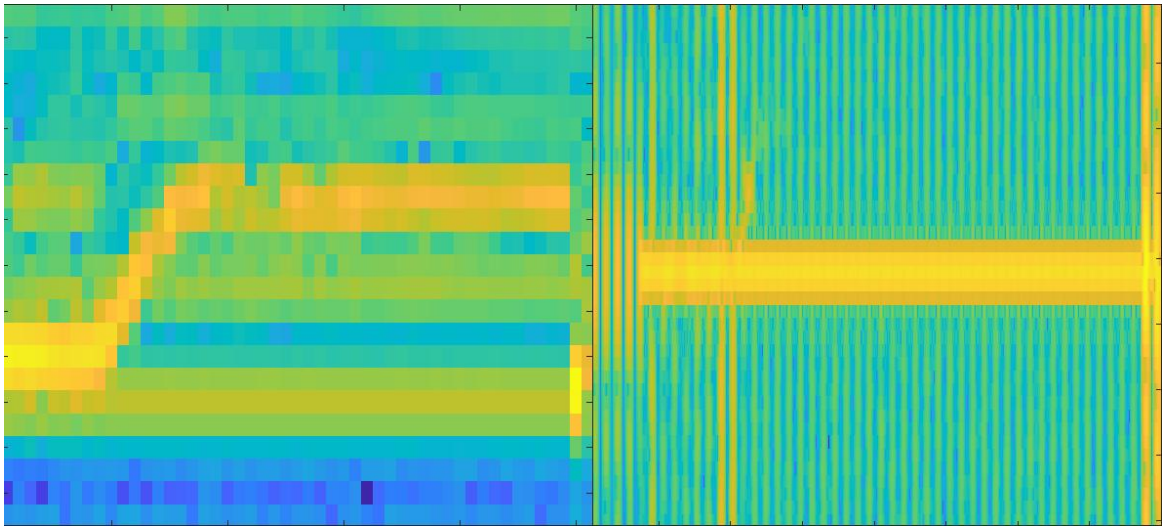


Figure 5.3: Combined image example of the pulling hand-gesture

The combined map presented in Figure 5.3 aligns with these expectations, as evident from the RTM side of the image. The target moves away from the vertical origin of the map, indicating that it is indeed moving away from the radar, which is consistent with the behaviour expected from a pulling hand gesture. However, there is some residual clutter present at the radar's original position, which should not be present. Regarding the DTM, the velocity experiences a spike before stabilising and remaining constant for the remaining data, as expected for a pulling hand gesture.

5.1.4 Pushing Hand-gesture

Concerning a pushing hand gesture, the expectations are similar to those for a pulling gesture. The range of the target is anticipated to change before staying constant, but this change should be opposite to that of the pulling gesture; the target should appear to move closer to the radar. In terms of the velocity of the target, a spike in velocity is expected on the opposite side of the DTM compared to the

pulling gesture. After this spike, the velocity should stabilise and remain constant for the duration of the DTM. Figure 5.4 below illustrates a sample combined image obtained representing the pushing hand gesture.

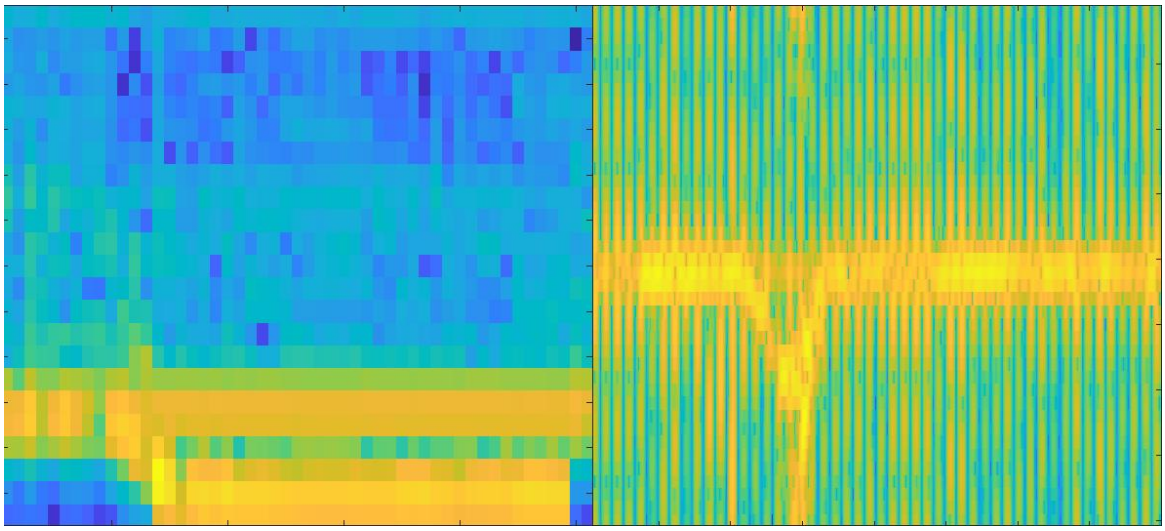


Figure 5.4: Combined image example of the pushing hand-gesture

Figure 5.4 adheres to these expectations. The target indeed moves closer to the radar, indicated by its approach to the vertical origin of the map. However, a similar issue is encountered as with the pulling hand gesture, where the radar continues to identify the target at its previous position throughout the remainder of the RTM. In the DTM, the velocity pattern aligns with expectations, with the velocity spike occurring on the opposite side compared to the pulling hand gesture. Following the spike, the velocity remains constant for the remainder of the DTM.

5.1.5 Patting Hand-gesture

The patting hand gesture is characterised as a prolonged hand gesture. It is anticipated that the radar's range will continuously fluctuate between different ranges. This is because the hand is moving closer and farther away from the radar as the hand gesture unfolds, which should manifest as a 'wave-like' pattern across the RTM. In terms of the target's velocity, it is expected to change from one side to the other throughout the entire duration of the gesture. Figure 5.5 below illustrates a sample combined image obtained representing the patting hand gesture.

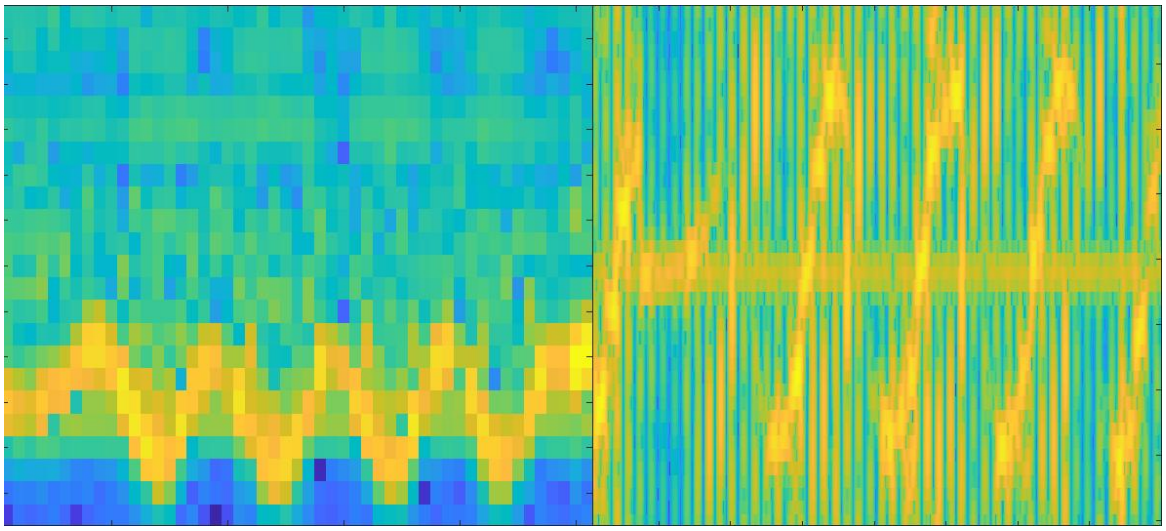


Figure 5.5: Combined image example of the patting hand-gesture

Figure 5.5 indeed conforms to these expectations. The RTM displays the target moving back and forth between different ranges, indicating that it is getting closer and then moving away from the radar over time. The DTM also aligns with the expectations as the velocity continuously shifts from one side to the other, signifying that the target is in motion.

5.1.6 Combined Maps Discussion

In general, the combined maps for each hand gesture closely resembles the expected outputs and display distinctive characteristics that are easily recognisable. However, is important to acknowledge that there are some issues related to the quality of the maps, particularly in terms of clutter and SNR. As observed with the pushing and pulling hand gestures, undesired signals that the radar picks up can be problematic. These unwanted signals might impact feature extraction within the maps and potentially reduce the performance of the classification algorithm. Furthermore, a significant amount of noise is still being detected by the radar, occasionally making it difficult to distinguish target information.

Nevertheless, despite these issues, the chosen hand gestures for this implementation can still be identified from the maps. This signifies that the processing pipeline has successfully met its requirement, which was to take raw radar data as input and generate maps containing information that can be extracted as features when fed into a machine learning algorithm. Consequently, these maps can be suitably used as inputs for the classification algorithm.

5.2 Classification Algorithm Results

5.2.1 Loss Function

Figure 5.6 below presents the loss function of the proposed CNN model. The loss function serves as a vital performance metric, offering insights into the degree of fitting of the model. It quantifies the difference between the predicted loss and the actual loss, monitored over the chosen number of training epochs. Overfitting, as discussed in Section 3.4.4, occurs when the model excels on the training data

but performs poorly on the testing data. Signs of overfitting can be identified using the loss curve by evaluating the training and validation losses. When the training loss is significantly smaller than the validation loss over numerous epochs, it often indicates overfitting. Overfitting is also evident when the validation loss increases while the training loss decreases, or when fluctuations are observed in the validation loss.

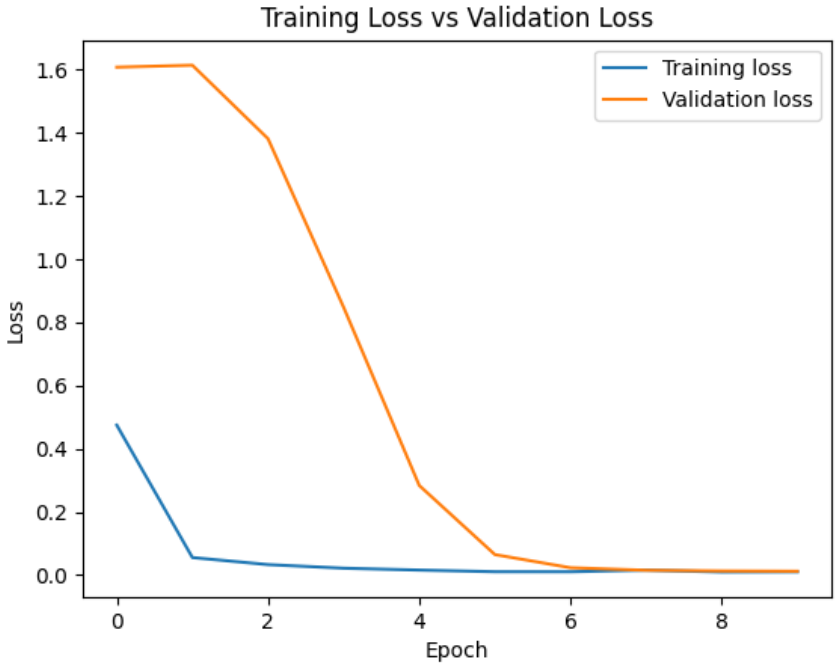


Figure 5.6: Loss curve of proposed CNN model

From the loss function displayed in Figure 5.6, it is evident that the proposed model is not subject to overfitting. Despite the validation loss starting higher than the training loss, the gap rapidly narrows as the number of epochs increases, becoming nearly identical after the 5th epoch. Furthermore, the validation loss consistently decreases over the epochs without any noticeable increases or fluctuations. This indicates that the model is robust and not overfitting. The loss curve demonstrates that the proposed CNN model generalises effectively and is able to achieve high performance on unseen data. Both the validation loss and the training loss reach zero after the 6th epoch, implying that the model attained a validation accuracy of 100% for the five hand gestures after the 6th epoch.

5.2.2 Confusion Matrix

The proposed CNN model was subsequently evaluated using unseen data from the testing set. The model's performance across each hand gesture on the test set was assessed through a confusion matrix, as depicted in Figure 5.7 below.

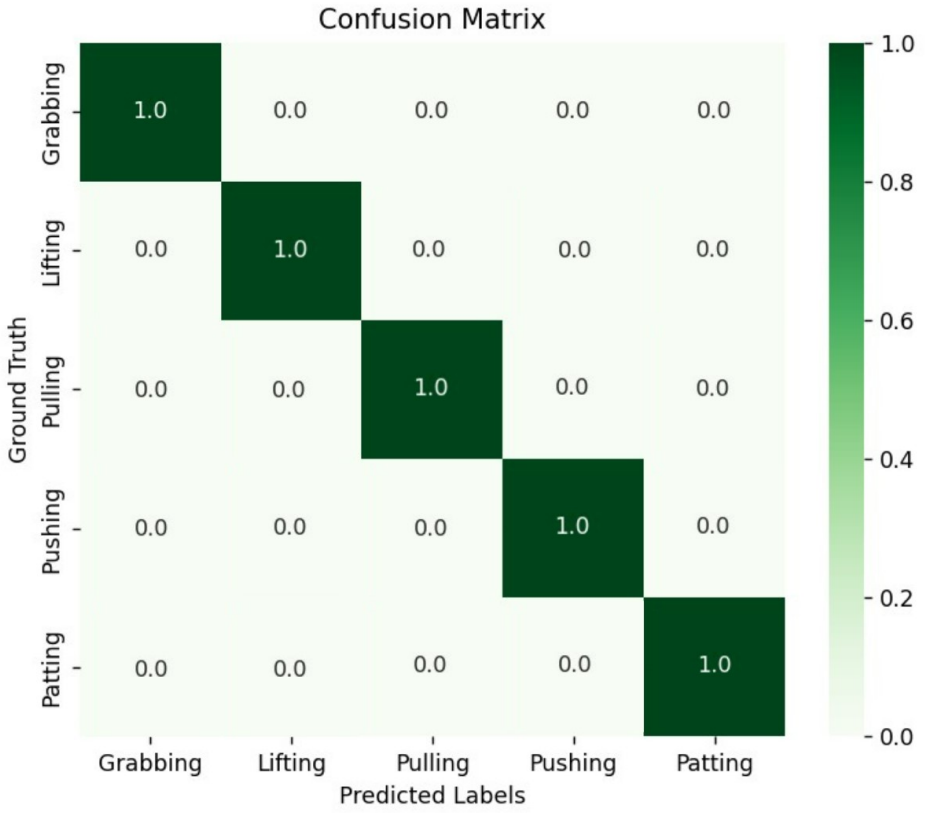


Figure 5.7: Confusion matrix of the proposed CNN model for the five hand gestures

In this confusion matrix, the labels in the rows correspond to the ground truth labels, while the columns represent the predicted labels generated by the proposed CNN model. Significantly, the confusion matrix reveals that the proposed model achieved 100% accuracy for all types of hand gestures.

5.3 Overall System Assessment

Given the differences in research aspects such as datasets, radar systems, processing pipelines, and machine learning algorithms, direct comparisons of results across different studies are meaningless. Consequently, a basic assessment is conducted to determine whether the obtained results align with the current state-of-the-art. This assessment aims to demonstrate whether the proposed system can achieve performance levels similar to previous studies.

In this evaluation, three prior research works serve as benchmarks for the proposed implementation: the studies conducted by Zhao et al. [49], Du et al. [53] and Wang et al. [42]. Notably, both Zhao et al. and Du et al. employed a multi-dimensional input approach for the machine learning algorithms, similar to the methodology of the proposed implementation. Meanwhile, Wang et al.'s approach utilised the same maps as the proposed implementation but employed a multi-channel machine learning algorithm. For a detailed breakdown of the unique characteristics of each of these studies, refer to the Related Works Section 2.3.

Research	Gestures Number	Input Maps	Classification Algorithm	Accuracy
[49]	3	RTM + DTM + ATM	Random Forrest Classifier	100%
[53]	10	RTM + DTM + ATM	CNN	91.34%
[42]	10	RTM + DTM	CNN	82.77%
Proposed System	5	RTM + DTM	CNN	100%

Table 5.1: Assessment of the proposed system with other systems in the current state-of-the-art

Table 5.1 provides an overview of the key characteristics and performance of each system. Despite the differences between these implementations, it is possible to gauge the performance of the proposed model by considering the achievements of past research. The performance of the model aligns with that of other studies. It would not be accurate to claim that the model outperforms the works of Du et al. and Wang et al., as these models were designed to recognise a larger range of hand gestures. The increased diversity in the types of hand gestures significantly influences the system's performance, as it complicates the classification process. However, it can be stated that the performance of the proposed model is on par with previous implementations.

5.4 Chapter 5 Summary

This chapter provided a comprehensive discussion of the proposed system's results. It first focused on the evaluation of the processing pipeline's performance and the output RTM and DTM. Generally, the output maps closely resembled the expected outcomes for different types of hand gestures. However, it is essential to note that certain maps exhibited suboptimal quality, characterised by a poor SNR and issues related to clutter. These undesirable signals have the potential to lead to performance challenges for a different dataset. Specifically, an increased number of hand gestures could complicate the classification algorithm's ability to distinguish between distinct features.

Despite these challenges, the proposed system achieved high accuracy results when applied to the five hand gestures designed. The fusion of various maps into a combined image, a technique previously observed only once in the work of Zhao et al. [49] in 2022, proved to be an effective way of providing features for the classification algorithm. Notably, the CNN model showed no signs of overfitting, as evident from the use of the loss function. It also demonstrated robust generalisation, ultimately achieving a perfect accuracy rate of 100% across all five predefined hand gestures on the new, unseen dataset.

Although direct comparisons may be limited due to differences in research methodologies, an assessment of the results obtained against prior research was done. This evaluation suggested that the results of the proposed system were in line with those achieved by similar studies in the field.

Chapter 6

Conclusions and Future Work

6.1 Report Summary

The objective of this report was to implement a HGR system using mmWave FMCW radar. This implementation required breaking down the task into several distinct stages, each presenting its unique engineering challenges and required choices and design considerations.

The report commenced in Chapter 2 with a comprehensive literature review, focusing on HGR using mmWave FMCW systems. This literature review laid the foundations for the entire report. It began by tracing the evolution of HGR technology to provide historical context to the study. This historical background highlighted the effectiveness of radar-based methods, particularly FMCW radar systems. Chapter 2 introduced the Google Soli Project as an early example of HGR using mmWave FMCW technology and also presented the work of Zhao et al. in 2022, which significantly influenced the methodology employed in this project. The literature review covered various approaches taken in the past, including the radar systems and their characteristics, the datasets employed, the mapping of raw radar data, and the significance of such maps, as well as the classification algorithms used. Additionally, this chapter included a section on related works, offering an overview of findings from prior research. The chapter concluded by discussing potential HGR applications, highlighting its recent rise in popularity, and addressing limitations identified in previous studies involving HGR using mmWave FMCW radar.

Chapter 3 continued with the essential theoretical background required for the implementation. It began by discussing fundamental radar principles and types of radar systems, progressing to the specifics of FMCW radar systems. The chapter then delved into the theoretical aspects of FMCW radar, discussing the signal description and the relevant equations needed for experimentation. Subsequently, the fundamentals of radar data processing were discussed, explaining the structure of raw radar data. The theory associated with generating key maps such as RDM, RTM and DTM was described, alongside crucial concepts like clutter and windowing. Chapter 3 concluded by introducing the theory behind classification algorithms, particularly CNN. This section detailed theory related to the model architecture and the training process.

Chapter 4 of the report detailed the design and implementation of the proposed system. It commenced by outlining the implementation approach and how the design process would unfold. The creation of a hand gesture dataset, encompassing five distinct hand gestures, was discussed, with a single participant (the author) gathering 150 samples for each gesture, yielding a total of 750 samples. The hardware used

for the implementation, comprising the AWR1843 radar with the DCA1000EVM, an Intel i7 NUC for the radar system, and an Acer Nitro 5 Intel Core i5-11400H laptop for signal processing and machine learning, was highlighted. The software selection process was elaborated, with MATLAB chosen for data processing and Python for the machine learning stage in conjunction with PyTorch. The chapter also delved into the data collection process with the radar configuration and parameter selection. It explained the processing pipeline, including reading raw radar data into MATLAB, data formatting for subsequent processing, map generation, strategies for addressing noise and clutter. The section also covered the process of saving the RTMs and DTMs for later use in the classification stage. The chapter concluded with the machine learning stage of the design, discussing the merging of the maps into a combined image, the splitting of the datasets, data augmentation techniques, the designing and training of the CNN model.

Chapter 5 presented an evaluation of the implemented system. It initiated by analysing the maps generated by the processing pipeline for each of the five hand gestures. Overall, the maps aligned with expectations, though some issues related to noise and clutter within the radar data were noted. The primary objective of the processing pipeline, which was to convert raw radar data into usable maps for machine learning, was successfully achieved. The chapter then displayed and discussed the results of the classification algorithm. This section displayed that the proposed model had no signs of overfitting and achieving a high accuracy rate of 100% for predictions across all five hand gestures on previously unseen data. Finally, the chapter concluded with a brief assessment of the proposed model against similar implementations in the current state-of-the-art, determining that the results obtained were in line with previous research findings.

6.2 Conclusions

The project successfully met all of its initially established objectives. Its primary goal was to develop and implement a HGR system using mmWave FMCW radar. The project achieved the design of a data processing pipeline capable of transforming raw radar data into a format suitable for HGR and as input for a machine learning algorithm to extract hand gestures.

As a proof of concept, a classification algorithm was designed and trained using the processed radar data obtained from the radar system. This classification algorithm demonstrated the ability to distinguish among five predefined hand gestures, achieving a remarkable classification accuracy of 100% when tested on previously unseen data. These results strongly indicate that the proposed system is on par with the past research in the current state-of-the-art.

6.3 Future Recommendations

While the proposed system currently functions adequately, there exist several areas for potential improvement in future work. The following section outlines recommendations for enhancing the system and suggests new features to be incorporated.

6.3.1 Additional Types of Hand-Gestures

Future work should explore the inclusion of additional types of hand gestures. A particular focus should be on integrating micro-gestures into the system, for example: zooming in, zooming out, clicking, or scrolling. The incorporation of new gestures would significantly improve the system's versatility, potentially making it suitable for integration into larger systems, particularly in various industrial applications. Furthermore, it is essential to increase the amount of data collected for each hand gesture, leading to a larger dataset. This will improve the model's generalisation capabilities. It is also advisable to diversify the dataset by collecting data from a more larger range of participants, encompassing both men and women, to enhance the model's robustness. Nonetheless, expanding the dataset comes with considerations, as discussed in Section 5.1.6, where noise and clutter negatively impacted the processing pipeline's quality. Therefore, the introduction of additional hand gestures would necessitate improvements to the processing pipeline.

6.3.2 Processing pipeline improvements

A clear recommendation involves enhancing the processing pipeline. One approach is to introduce additional radar maps. The incorporation of an ATM within the processing pipeline could facilitate distinguishing hand gestures that closely resemble each other, based on the angle at which they are from the radar system. The implementation of an angle-based feature over time would be feasible only if the radar system possesses multiple receiver antennas. In the event of introducing more hand gestures, it is important to acknowledge that the classification model's accuracy may decline. However, the addition of a third processing map would enable the creation of a larger combined image with more distinguishing features. This could potentially offset the accuracy decrease by improving the performance of the classification algorithm.

Another critical aspect requiring improvement within the processing pipeline is the mitigation of challenges posed by clutter within the system. One suggested solution is the incorporation of a background modelling technique to effectively differentiate dynamic targets from static clutter. An algorithm well-suited for this purpose could be the Greedy bilateral smoothing (GreBsmo) algorithm, which has been used in previous research. Furthermore, addressing the issue of noise within the system could involve the implementation of a Constant False Alarm Rate (CFAR) method within the processing pipeline. A CFAR method would allow the system to detect hand gestures against background noise or clutter, ultimately leading to enhanced processing pipeline performance and improved overall system efficiency.

6.3.3 Real-Time Implementation

In its current form, the proposed system is a prototype and lacks real-time functionality. In the future, it would be beneficial to develop a system capable of real-time HGR. This enhancement would significantly increase the system's versatility and potential applications in HCI devices. An option would be to integrate the system into a smart terminal, allowing it to collect data, process it, and make predictions using the classification algorithm in real-time. However, it is important to note that a real-time implementation would necessitate additional computational considerations and requirements, potentially requiring the redesign of certain system components.

Chapter 7

Appendix

7.1 GitHub

A link to the GitHub for this project which has all the implementaion done:

https://github.com/EthanMeknassi/EEE4022S

7.2 Literature Review

Methods and Data	Hand Gesture Classification						
	PS	PL	LF	RG	P-P	U-G	Average
KNN+RTM	57	65	89	88	100	98	82.83
KNN+DTM	100	77	88	85	61	100	85.17
KNN+ATM	78	80	97	93	79	65	82.00
DTW+RTM	92	76	93	77	100	99	89.50
DTW+DTM	94	92	98	87	69	99	89.83
DTW+ATM	89	90	97	91	83	81	88.50
DTW+RTM+DTM	98	92	93	96	93	95	94.50
FDTW	**99**	**92**	**94**	**98**	**95**	**97**	**95.83**

Figure 7.1: Wang et al.'s comparison of the different input features and classification algorithm [60]

Methods	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)	(o)	(p)	Avg.
2D-CNN (RTM+DTM)	92.47	91.95	91.00	92.67	97.33	93.19	92.34	90.23	94.42	90.35	96.48	91.44	88.74	97.38	90.93	96.10	92.93
2D-CNN (RTM+DTM +ATM+ETM)[20]	99.06	99.17	99.35	99.22	99.3	99.28	97.77	99.17	99.17	99.26	99.39	99.96	94.18	99.17	99.46	99.15	98.87
3D-CNN (MPCA)[22]	95.21	95.18	95.99	95.08	97.87	95.47	96.78	97.14	97.64	94.56	99.02	94.46	89.00	94.96	93.68	95.16	95.45
3D-CNN +LSTM[23]	98.38	98.38	98.44	99.54	99.14	100.00	99.54	97.83	98.48	97.87	99.57	99.02	92.38	97.34	98.36	95.49	98.11
2D-CNN (Multi-feature encoder)[23]	98.35	98.35	98.40	99.45	98.92	100.0	99.45	97.80	98.38	97.85	99.45	98.95	92.34	97.27	98.28	95.40	98.04
S3D (5D feature cubes)	98.51	99.20	99.70	99.55	99.42	99.14	97.76	99.55	99.50	99.69	99.46	98.80	93.65	99.47	99.04	98.50	98.80
S3D +STDC **(ours)**	99.51	99.70	99.27	99.42	99.74	99.62	98.04	99.47	99.60	99.90	99.86	99.35	93.98	99.70	99.35	99.55	99.12
S3D +ASTCAC **(ours)**	100.0	99.58	98.88	99.39	99.87	97.27	97.27	99.32	99.92	99.65	98.88	99.71	93.68	99.48	99.94	99.79	99.01
S3D +STDC +ASTCAC **(ours)**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**98.11**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**94.51**	**100.0**	**100.0**	**100.0**	**99.53**

Figure 7.2: Dong et al.'s comparison of the different classification algorithms' accuracies [51]

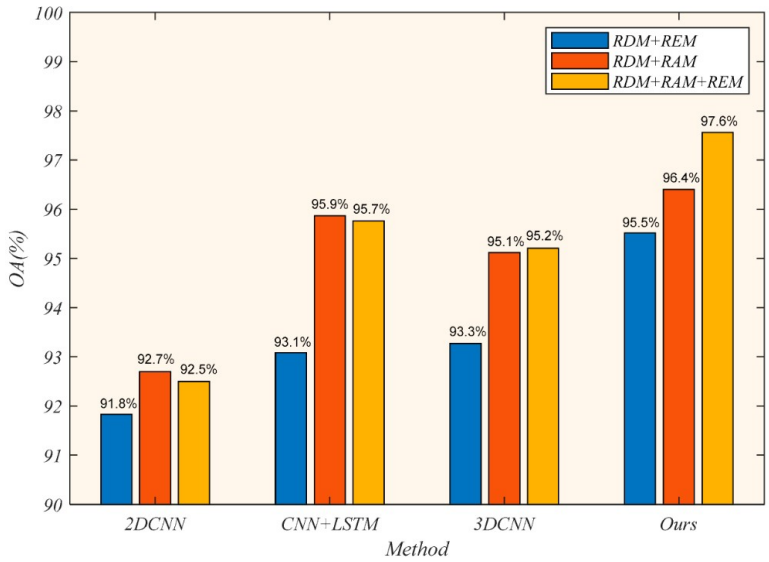


Figure 7.3: Zheng et al.'s comparison of the different classification models for different input feature [46]

Bibliography

[1] T. G. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill, "A HAND GESTURE INTERFACE DEVICE," http://netzspannung.org/cat/servlet/CatServlet/$files/228648/DataGlove+CHI+1987.pdf, accessed: 2023-9-18.

[2] H. P. Gupta, H. S. Chudgar, S. Mukherjee, T. Dutta, and K. Sharma, "A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors," *IEEE Sens. J.*, vol. 16, no. 16, pp. 6425–6432, 2016.

[3] J. Younas, S. Narayan, and P. Lukowicz, "Air-Writing segmentation using a single IMU-based system," in *2023 19th International Conference on Intelligent Environments (IE)*. IEEE, 2023, pp. 1–6.

[4] Y. Gao, S. Zeng, J. Zhao, W. Liu, and W. Dong, "AirText: One-handed text entry in the air for COTS smartwatches," *IEEE Trans. Mob. Comput.*, vol. 22, no. 5, pp. 2506–2519, 2023.

[5] Z. Lu, X. Chen, Q. Li, X. Zhang, and P. Zhou, "A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices," *IEEE Trans. Hum. Mach. Syst.*, vol. 44, no. 2, pp. 293–299, 2014.

[6] G. Singh, A. Nelson, R. Robucci, C. Patel, and N. Banerjee, "Inviz: Low-power personalized gesture recognition using wearable textile capacitive sensor arrays," in *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2015, pp. 198–206.

[7] G. Rogez, J. S. Supancic, III, and D. R. I. Rhone-Alpes, "Understanding everyday hands in action from RGB-D images," https://openaccess.thecvf.com/content_iccv_2015/papers/Rogez_Understanding_Everyday_Hands_ICCV_2015_paper.pdf, accessed: 2023-9-18.

[8] C. Zimmermann and T. Brox, "Learning to estimate 3D hand pose from single RGB images," https://openaccess.thecvf.com/content_ICCV_2017/papers/Zimmermann_Learning_to_Estimate_ICCV_2017_paper.pdf, accessed: 2023-9-18.

[9] E. Ohn-Bar and M. M. Trivedi, "Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2368–2377, 2014.

[10] A. Zabatani, V. Surazhsky, E. Sperling, S. B. Moshe, O. Menashe, D. H. Silver, Z. Karni, A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Intel® RealSense™ SR300 coded light depth camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2333–2345, 2020.

[11] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013.

[12] H. Abdelnasser, M. Youssef, and K. A. Harras, "WiGest: A ubiquitous WiFi-based gesture recognition system," in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 1472–1480.

[13] Z. Tian, J. Wang, X. Yang, and M. Zhou, "WiCatch: A WI-fi based hand gesture recognition system," *IEEE Access*, vol. 6, pp. 16 911–16 923, 2018.

[14] M. Al-qaness and F. Li, "WiGeR: WiFi-based gesture recognition system," *ISPRS Int. J. Geoinf.*, vol. 5, no. 6, p. 92, 2016.

[15] M. Yang, H. Zhu, R. Zhu, F. Wu, L. Yin, and Y. Yang, "WiTransformer: A novel robust gesture recognition sensing model with WiFi," *Sensors (Basel)*, vol. 23, no. 5, p. 2612, 2023.

[16] Z. Xia, Y. Luomei, C. Zhou, and F. Xu, "Multidimensional feature representation and learning for robust hand-gesture recognition on commercial millimeter-wave radar," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 4749–4764, 2021.

[17] T. Fan, C. Ma, Z. Gu, Q. Lv, J. Chen, D. Ye, J. Huangfu, Y. Sun, C. Li, and L. Ran, "Wireless hand gesture recognition based on continuous-wave doppler radar sensors," *IEEE Trans. Microw. Theory Tech.*, vol. 64, no. 11, pp. 4012–4020, 2016.

[18] S. Skaria, A. Al-Hourani, M. Lech, and R. J. Evans, "Hand-gesture recognition using two-antenna doppler radar with deep convolutional neural networks," *IEEE Sens. J.*, vol. 19, no. 8, pp. 3041–3048, 2019.

[19] S. Skaria, A. Al-Hourani, and R. J. Evans, "Deep-learning methods for hand-gesture recognition using ultra-wideband radar," *IEEE Access*, vol. 8, pp. 203 580–203 590, 2020.

[20] S. Mekruksavanich, P. Jantawong, D. Tancharoen, and A. Jitpattanakul, "A convolutional neural network for ultra-wideband radar-based hand gesture recognition," in *2023 International Technical Conference on Circuits/Systems, Computers, and Communications (ITC-CSCC)*. IEEE, 2023, pp. 1–4.

[21] J. Lien, N. Gillian, M. Emre Karagozler, P. Amihood, C. Schwesig, E. Olson, H. Raja, and I. Poupyrev, "Soli: Ubiquitous gesture sensing with millimeter wave radar."

[22] G. Tang, T. Wu, and C. Li, "Dynamic gesture recognition based on FMCW millimeter wave radar: Review of methodologies and results," *Sensors (Basel)*, vol. 23, no. 17, p. 7478, 2023.

[23] N. Magrofuoco, J.-L. Pérez-Medina, P. Roselli, J. Vanderdonckt, and S. Villarreal, "Eliciting contact-based and contactless gestures with radar-based sensors," vol. 7, pp. 176 982–176 997, conference Name: IEEE Access. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8890919

[24] Y.-C. Jhaung, Y.-M. Lin, C. Zha, J.-S. Leu, and M. Köppen, "Implementing a hand gesture recognition system based on range-doppler map," *Sensors (Basel)*, vol. 22, no. 11, p. 4260, 2022.

[25] Y. Li, C. Gu, and J. Mao, "4-d gesture sensing using reconfigurable virtual array based on a 60-ghz fmcw mimo radar sensor," *IEEE Transactions on Microwave Theory and Techniques*, vol. 70, no. 7, pp. 3652–3665, 2022.

[26] A. Ali, P. Parida, V. Va, S. Ni, K. N. Nguyen, B. L. Ng, and J. C. Zhang, "End-to-end dynamic gesture recognition using mmwave radar," *IEEE Access*, vol. 10, pp. 88 692–88 706, 2022.

[27] Z. Zhang, Z. Tian, and M. Zhou, "Latern: Dynamic continuous hand gesture recognition using fmcw radar sensor," *IEEE Sensors Journal*, vol. 18, no. 8, pp. 3278–3289, 2018.

[28] J.-W. Choi, S.-J. Ryu, and J.-H. Kim, "Short-range radar based real-time hand gesture recognition using lstm encoder," *IEEE Access*, vol. 7, pp. 33 610–33 618, 2019.

[29] J.-T. Yu, L. Yen, and P.-H. Tseng, "mmwave radar-based hand gesture recognition using range-angle image," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.

[30] D. Rodrigues and C. Li, "Hand gesture recognition using fmcw radar in multi-person scenario," in *2021 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNeT)*, 2021, pp. 50–52.

[31] J. Lien, N. Gillian, M. Emre Karagozler, P. Amihood, C. Schwesig, E. Olson, H. Raja, and I. Poupyrev, "Soli: Ubiquitous gesture sensing with millimeter wave radar."

[32] "Search," https://www.kaggle.com/search?q=hand+gesture, accessed: 2023-10-23.

[33] K. Ramasubramanian, "Moving from legacy 24 GHz to state-of-the-art 77 GHz radar," https://www.ti.com/lit/wp/spry312/spry312.pdf?ts=1698049595272&ref_url=https%253A%252F%252Fwww.google.com%252F#:~:text=The%20velocity%20resolution%20and%20accuracy,resolution%20and%20accuracy%20by%203x., accessed: 2023-10-23.

[34] L. Qu, H. Wu, T. Yang, L. Zhang, and Y. Sun, "Dynamic hand gesture classification based on multichannel radar using multistream fusion 1-D convolutional neural network," *IEEE Sens. J.*, vol. 22, no. 24, pp. 24 083–24 093, 2022.

[35] S. Hazra and A. Santra, "Robust gesture recognition using millimetric-wave radar system," *IEEE Sensors Letters*, vol. 2, no. 4, pp. 1–4, 2018.

[36] M. A. Simão, O. Gibaru, and P. Neto, "Online recognition of incomplete gesture data to interface collaborative robots," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9372–9382, 2019.

[37] G. Zhang, S. Lan, K. Zhang, and L. Ye, "Temporal-range-doppler features interpretation and recognition of hand gestures using mmw fmcw radar sensors," in *2020 14th European Conference on Antennas and Propagation (EuCAP)*, 2020, pp. 1–4.

[38] M. S. S, M. Padavala, and J. Valarmathi, "Hand gesture recognition based application using 60 ghz fmcw mimo radar," in *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)*, 2023, pp. 1–6.

[39] B. Jin, X. Ma, Z. Zhang, Z. Lian, and B. Wang, "Interference-robust millimeter-wave radar-based dynamic hand gesture recognition using 2d cnn-transformer networks," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[40] 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.

[41] W. Jiang, Y. Ren, Y. Liu, Z. Wang, and X. Wang, "Recognition of dynamic hand gesture based on mm-wave fmcw radar micro-doppler signatures," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 4905–4909.

[42] Y. Wang, S. Wang, M. Zhou, Q. Jiang, and Z. Tian, "Ts-i3d based hand gesture recognition method with radar sensor," *IEEE Access*, vol. 7, pp. 22 902–22 913, 2019.

[43] Y. Wang, A. Ren, M. Zhou, W. Wang, and X. Yang, "A novel detection and recognition method for continuous hand gesture using fmcw radar," *IEEE Access*, vol. 8, pp. 167 264–167 275, 2020.

[44] M. Arsalan, A. Santra, and V. Issakov, "Radarsnn: A resource efficient gesture sensing system based on mm-wave radar," *IEEE Transactions on Microwave Theory and Techniques*, vol. 70, no. 4, pp. 2451–2461, 2022.

[45] Y. Wang, D. Wang, Y. Fu, D. Yao, L. Xie, and M. Zhou, "Multi-hand gesture recognition using automotive FMCW radar sensor," *Remote Sens. (Basel)*, vol. 14, no. 10, p. 2374, 2022.

[46] L. Zheng, J. Bai, X. Zhu, L. Huang, C. Shan, Q. Wu, and L. Zhang, "Dynamic hand gesture recognition in in-vehicle environment based on FMCW radar and transformer," *Sensors (Basel)*, vol. 21, no. 19, p. 6368, 2021.

[47] Y. Zhao, V. Sark, M. Krstic, and E. Grass, "Low complexity radar gesture recognition using synthetic training data," *Sensors (Basel)*, vol. 23, no. 1, p. 308, 2022.

[48] Z. Yang and X. Zheng, "Hand gesture recognition based on trajectories features and computation-efficient reused lstm network," *IEEE Sensors Journal*, vol. 21, no. 15, pp. 16 945–16 960, 2021.

[49] Y. Zhao, V. Sark, M. Krstic, and E. Grass, "Novel approach for gesture recognition using mmwave fmcw radar," in *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, 2022, pp. 1–6.

[50] ——, "Novel approach for gesture recognition using mmwave FMCW RADAR," in *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*. IEEE, 2022, pp. 1–6.

[51] X. Dong, Z. Zhao, Y. Wang, T. Zeng, J. Wang, and Y. Sui, "Fmcw radar-based hand gesture recognition using spatiotemporal deformable and context-aware convolutional 5-d feature representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2022.

[52] X. Shen, H. Zheng, X. Feng, and J. Hu, "Ml-hgr-net: A meta-learning network for fmcw radar based hand gesture recognition," *IEEE Sensors Journal*, vol. 22, no. 11, pp. 10 808–10 817, 2022.

[53] C. Du, L. Zhang, X. Sun, J. Wang, and J. Sheng, "Enhanced multi-channel feature synthesis for hand gesture recognition based on cnn with a channel and spatial attention mechanism," *IEEE Access*, vol. 8, pp. 144 610–144 620, 2020.

[54] J. S. Suh, S. Ryu, B. Han, J. Choi, J.-H. Kim, and S. Hong, "24 ghz fmcw radar system for real-time hand gesture recognition using lstm," in *2018 Asia-Pacific Microwave Conference (APMC)*, 2018, pp. 860–862.

[55] K. A. Smith, C. Csech, D. Murdoch, and G. Shaker, "Gesture recognition using mm-wave sensor for human-car interface," *IEEE Sensors Letters*, vol. 2, no. 2, pp. 1–4, 2018.

[56] M. Q. Nguyen, A. Flores-Nigaglioni, and C. Li, "Range-gating technology for millimeter-wave radar remote gesture control in iot applications," in *2018 IEEE MTT-S International Wireless Symposium (IWS)*, 2018, pp. 1–4.

[57] Z. Li, Z. Lei, A. Yan, E. Solovey, and K. Pahlavan, "Thumouse: A micro-gesture cursor input through mmwave radar-based interaction," in *2020 IEEE International Conference on Consumer Electronics (ICCE)*, 2020, pp. 1–9.

[58] M. R. Abid, E. M. Petriu, and E. Amjadian, "Dynamic sign language recognition for smart home interactive application using stochastic linear formal grammar," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 3, pp. 596–605, 2015.

[59] Y. Qifan, T. Hao, Z. Xuebing, L. Yin, and Z. Sanfeng, "Dolphin: Ultrasonic-based gesture recognition on smartphone platform," in *2014 IEEE 17th International Conference on Computational Science and Engineering*, 2014, pp. 1461–1468.

[60] Z. Wang, F. Liu, X. Li, M. Ma, X. Feng, and Y. Guo, "A survey of hand gesture recognition based on FMCW radar," in *Proceedings of the 8th International Conference on Communication and Information Processing*. New York, NY, USA: ACM, 2022.

[61] Z. Zhang, Z. Tian, Y. Zhang, M. Zhou, and B. Wang, "u-deephand: Fmcw radar-based unsupervised hand gesture feature learning using deep convolutional auto-encoder network," *IEEE Sensors Journal*, vol. 19, no. 16, pp. 6811–6821, 2019.

[62] M. A. Richards, J. A. Scheer, and W. A. Holm, "Principles of modern radar," https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=8da4c3b56c227db379e0d454a0285f84fba8ee1f, accessed: 2023-10-24.

[63] M. Skolnik, "Radar handbook," https://ftp.idu.ac.id/wp-content/uploads/ebook/tdg/ADNVANCED%20MILITARY%20PLATFORM%20DESIGN/Radar%20Handbook.pdf, accessed: 2023-10-24.

[64] L. Senigagliesi, G. Ciattaglia, A. De Santis, and E. Gambi, "People walking classification using automotive radar," *Electronics (Basel)*, vol. 9, no. 4, p. 588, 2020.

[65] V. Dham, "Programming chirp parameters in TI radar devices," https://www.ti.com/lit/an/swra553a/swra553a.pdf?ts=1697638455692&ref_url=https%253A%252F%252Fwww.google.com%252F, accessed: 2023-10-18.

[66] H. Park, M. Kim, Y. Jung, and S. Lee, "Method for improving range resolution of indoor FMCW radar systems using DNN," *Sensors (Basel)*, vol. 22, no. 21, p. 8461, 2022.

[67] "Radar clutter," https://skybrary.aero/articles/radar-clutter, accessed: 2023-10-24.

[68] Y. Roshni, "Radar clutter," https://electronicsdesk.com/radar-clutter.html, Oct. 2019, accessed: 2023-10-24.

[69] "Understanding FFTs and windowing overview," https://download.ni.com/evaluation/pxi/Understanding%20FFTs%20and%20Windowing.pdf, accessed: 2023-10-24.

[70] E. Hughes, "Windowing functions in radar technology," https://www.skyradar.com/blog/windowing-functions-in-radar-technology, Mar. 2021, accessed: 2023-10-24.

[71] "No title," https://www.deeplearningbook.org/contents/convnets.html, accessed: 2023-10-22.

[72] M. Mandal, "Introduction to convolutional neural networks (CNN)," https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/, May 2021, accessed: 2023-10-22.

[73] https://www.analog.com/media/en/technical-documentation/dsp-book/dsp_book_ch6.pdf, accessed: 2023-10-22.

[74] Raycad, "Convolutional neural network (CNN)," https://medium.com/@raycad.seedotech/convolutional-neural-network-cnn-8d1908c010ab, Nov. 2017, accessed: 2023-10-22.

[75] S. Bhardwaj, "Convolutional neural networks : Understand the basics," https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-understand-the-basics/, May 2021, accessed: 2023-10-22.

[76] X. Zhou, H. Liu, C. Shi, and J. Liu, "The basics of deep learning," in *Deep Learning on Edge Computing Devices*. Elsevier, 2022, pp. 19–36.

[77] https://www.baeldung.com/cs/batch-normalization-cnn, accessed: 2023-10-22.

[78] M. Arham, "Diving into the pool: Unraveling the magic of CNN pooling layers," https://www.kdnuggets.com/diving-into-the-pool-unraveling-the-magic-of-cnn-pooling-layers, accessed: 2023-10-22.

[79] M. S. Ali, "Flattening CNN layers for neural network and basic concepts," https://medium.com/@muhammadshoaibali/flattening-cnn-layers-for-neural-network-694a232eda6a, Jun. 2022, accessed: 2023-10-22.

[80] S. Sharma, "Activation functions in neural networks," https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6, Sep. 2017, accessed: 2023-10-22.

[81] https://www.researchgate.net/figure/The-plot-of-the-ReLU-function_fig5_335540811, accessed: 2023-10-22.

[82] P. Baheti, "Train test validation split: How to & best practices [2023]," https://www.v7labs.com/blog/train-validation-test-set, Apr. 2023, accessed: 2023-10-22.

[83] O. Dreessen, "Training convolutional neural networks: What is machine learning?—part 2," https://www.analog.com/en/analog-dialogue/articles/training-convolutional-neural-networks-what-is-machine-learning-part-2.html, Mar. 2023, accessed: 2023-10-23.

[84] Rendyk, "Tuning the hyperparameters and layers of neural network deep learning," https://www.analyticsvidhya.com/blog/2021/05/tuning-the-hyperparameters-and-layers-of-neural-network-deep-learning/, May 2021, accessed: 2023-10-23.

[85] "Generalization in neural networks," https://www.kdnuggets.com/2019/11/generalization-neural-networks.html, accessed: 2023-10-23.

[86] B. Tara, "Improving generalizability of CNN models," https://medium.com/rescon-ai/improving-generalizability-of-cnn-models-3ca70bca4fec, Sep. 2021, accessed: 2023-10-23.

[87] "AWR1843," https://www.ti.com/product/AWR1843, accessed: 2023-10-18.

[88] "DCA1000EVM," https://www.ti.com/product/DCA1000EVM/part-details/DCA1000EVM, accessed: 2023-10-18.

[89] https://dev.ti.com/gallery/view/mmwave/mmWave_Demo_Visualizer/ver/2.1.0/, accessed: 2023-10-18.

[90] "Mmwave radar device ADC raw data capture," https://www.ti.com/lit/an/swra581b/swra581b.pdf?ts=1696948288095&ref_url=https%253A%252F%252Fwww.google.com%252F#:~:text=The%20DCA1000%20captured%20data%20samples,LVDS%20lane%20will%20be%20stored., 2017, accessed: 2023-10-18.

[91] L. M. G. M. S. P. P. A. A. M. Azraa Vally, Nicholas Bowden and A. Patel, "M2s2: A multi-modal sensor system for remote animal motion capture in the wild," *Unpublished*.

[92] "ConvNet shape calculator," https://madebyollin.github.io/convnet-calculator/, accessed: 2023-10-22.

UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA · UNIVERSITEIT VAN KAAPSTAD

2023/09/29

EBE/00385/2023

RE: Research Ethics Committee Project Approval Letter

Dear Ethan Meknassi,

Your application for ethics review of your project titled

Target Motion Reconstruction using Frequency Modulated Continuous Wave radar for Hand Gestures

has been reviewed and evaluated by the

Engineering & Built Environment Committee.

You may proceed with your research project titled:

Target Motion Reconstruction using Frequency Modulated Continuous Wave radar for Hand Gestures

Please note that should:

(i) any serious or adverse effects to participants occur and/or,

(ii) aspect(s) of your current project change and/or

(iii) any unforeseen events that might affect continued ethical acceptability of the project occur then you should immediately report this to the approving REC. You may be required to submit an amendment to this application, in order to determine whether the changed aspects increase the ethical risks of your project.

Based on the information supplied your application has been successful and is approved.

Please note the following additional conditions associated with this approval:

(i) Noted that all was in place for a positive review outcome by the 3rd attempt.

Regards,

Engineering & Built Environment Committee.