University of Glasgow | School of Computing Science

Techniques for Subtle Mid-Air Gestural Interaction Using mmWave Radar

Anith Manu Ravindran

Submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

January 2025

To Amma and Acha...

Abstract

Users need to be able to interact with mid-air gesture systems in ways that are efficient, precise, and socially acceptable. Subtle mid-air micro gestures can provide low-effort and discreet ways of interaction. This thesis contributes techniques for recognizing and utilizing subtle mid-air gestures with millimeter wave radars, a rapidly emerging sensing technology in human-computer interaction.

The first contribution focused on the problem of addressing a system. By analyzing the frequency components of various hand motions, subtle activation gestures were identified which produced high-frequency signals through deliberate, rhythmic movements. A novel activation gesture recognition pipeline was then developed using frequency analysis to recognize these gestures and ignore incidental hand motions. Tested across three types of sensors, the pipeline demonstrated robust performance in recognizing subtle high-frequency activation gestures and producing zero false activations for broad hand motions. Further improvements were also explored to enhance robustness to reduce false activations during activities like typing, writing, and phone usage.

The second contribution focused on recognition of subtle gestures from mmWave radar data using deep learning. A new dataset was developed, capturing the temporal dynamics and motion patterns of 10 different subtle gestures from 8 users with a mmWave radar. Multiple neural network architectures were trained and evaluated using the dataset, achieving a high recognition accuracy of 90%. The results demonstrated that hybrid neural networks combining convolutional and recurrent layers can effectively recognize subtle gestures from mmWave radar signals and generalize across different users.

The final contribution progressed from offline evaluations to practical, real-time assessments. The neural network models were integrated into prototype applications that enabled real-time subtle gesture interactions for tasks such as selecting photos and adjusting media playback. A user study demonstrated significant improvements in task completion, accuracy, and user experience compared to traditional macro gestures. The findings suggest that subtle gestural interaction, enabled by mmWave radar sensors, signal processing, and deep learning, can significantly enhance usability of virtual interfaces.

Contents

List of Tables

List of Figures

Acknowledgements

The research in this thesis was partly funded by the Google Advanced Technology and Projects group, who also provided the radar sensor used in this work.

First, I would like to sincerely thank Prof. Roderick Murray-Smith — I couldn't have asked for a better supervisor, guiding me not just through my PhD but all the way from my undergraduate studies.

I would also like to thank everyone, past and present, in the Inference, Dynamics and Interaction research group for creating such a friendly and supportive environment. Special thanks to Dr. Chaitanya Kaul, my secondary supervisor, for his words of encouragement and support, and to Andrew Ramsay for his incredible technical help with code, sensor setups, and all sorts of troubleshooting.

I would like to thank my thesis committee — Euan Freeman and Aaron Quigley — for the engaging discussions during my viva and for their valuable feedback, which greatly helped strengthen this thesis.

Finally, to my parents, I am so grateful for their sacrifices, unconditional support, and constant encouragement throughout my life, without which none of this would have been possible.

Declaration

I declare that this thesis represents my own work and has not been submitted for the award of any other degree. All sources of information have been properly cited and acknowledged where appropriate.

1 Introduction

1.1 Motivation

A mid-air gesture is a form of input that allow users to control or communicate with a smart device (e.g. smartwatch, smart TV, smart speaker) by making defined hand movements in the space around a sensing device, without needing any physical contact. This type of interaction technique has been extensively studied since the 1980s and continues to garner increasing interest, driven primarily by ongoing advancements in sensor technologies that make mid-air gestures more practical and effective across a wide range of applications.

1.1.1 Usability Challenges With Macro-Gestures

Interaction with smart devices occurs frequently across various settings, ranging from private (e.g., homes, personal vehicles) to public spaces (e.g., offices, libraries, public transport). Traditionally, mid-air gestural interaction has relied on large hand and body movements. These are referred to as macro-movements or macro-gestures. While these gestures are suitable in applications like gaming (e.g., Wii or Xbox Kinect), they pose usability challenges in other day-to-day interactions with devices. To understand these challenges, consider the following scenario:

Preethi loves binge-watching TV shows and often uses wave and swipe mid-air gestures to activate gesture recognition on her TV and navigate through apps and shows. However, these gestures quickly become tiring. Repeatedly lifting and waving her arms to activate the TV and swiping to scroll through programs leads to physical discomfort. To make matters worse, her gestures are sometimes mistakenly detected by other smart devices in the room, causing the smart lights to turn on or the smart speaker to play music. When friends come over to watch a movie, Preethi feels self-conscious about using large macro-gestures in front of them. The exaggerated movements make her worry about looking awkward, so she opts to use the TV remote instead.

Preethi encounters a few significant problems. Firstly, the repeated use of macro-gestures to control her TV leads to physical discomfort; to scroll through shows or episodes, she has to frequently use swipe gestures. The frequent arm movements becomes tiresome, causing physical fatigue.

Her second problem is that when she waves her hand to activate the TV, the gesture is unintentionally detected by other smart devices in the room. This illustrates the *Midas Touch* problem, where the continuous monitoring for gestures by multiple devices causes any recognized signal to be considered as intentional interaction, even if it was meant for another device.

The last problem she encounters is related to social acceptability. She is influenced by her social environment and the presence of others, which affects her willingness to engage in certain types of interactions. Performing large gestures could make them appear socially awkward or out of place. Additionally, such gestures may be physically unfeasible in certain situations, such as when sitting on a couch with friends or in other close quarters where people are seated shoulder to shoulder.

The systems developed in this thesis aim to address these challenges using subtle mid-air gestures. The term *subtle gesture* can take on different meanings depending on context.

For example, even a large macro-gesture can appear subtle if socially inconspicuous—such as swiping the cuff of a smart jacket, which to others may resemble a natural adjustment. In such cases, subtlety arises from social camouflage rather than physical scale. However, in HCI literature, *subtle gestures* more commonly refer to low-effort, compact hand or finger motions—closely aligned with what are known as *micro-gestures*. Micro-gestures refer to small-scale movements performed within a limited interaction space, typically spanning only a few centimeters [92, 91, 10]. They primarily involve smaller muscle groups, such as those controlling individual fingers. Such gestures are subtle because they are visually discreet, non-intrusive, and designed to blend naturally into everyday activities. While not all macro-gestures are subtle, all micro-gestures are, by nature, subtle. This thesis uses the term *subtle gesture* to refer to these low-amplitude hand and finger motions that are spatially compact, minimally demanding in terms of physical exertion, and socially unobtrusive.

1.1.2 The Rise of Radars in Gesture Sensing

Gesture detection has traditionally relied on vision-based sensing (e.g., Kinect, RealSense, Optitrack). While these sensors play a vital role in gestural interaction, they also come with certain limitations. For example, in Preethi's scenario, assume the TV is equipped with a camera for gesture control. In a dimly lit or dark room, the camera might struggle to detect her hand due to its reliance on proper lighting conditions. Another significant concern is privacy. While some users may appreciate the convenience of hands-free interaction, others may be wary of devices that track them visually. For these privacy-conscious users, the idea of having a camera that constantly monitors them raises concerns over surveillance and data security.

Recent efforts have led to the development of low-cost, miniaturized radars that possess properties to overcome many of the challenges posed by vision-based sensors. Radars use radio waves, a type of electromagnetic signal, to detect the range and speed of

objects [73]. Unlike vision-based sensors, these signals are unaffected by poor lighting or atmospheric conditions and can even penetrate through surfaces and objects, allowing for more consistent and reliable gesture detection in various environments [94]. Radars also offer a more privacy-preserving alternative to traditional vision-based systems. Unlike cameras, which capture detailed visual imagery, the radars systems used in this work sense only motion-related information such as distance, speed, and direction of movement. This makes them far less intrusive, as no visual details of the user or their surroundings are recorded.

The most significant property of radars relevant for this research is their sensitivity to subtle motions. Radars, specifically millimeter-wave (mmWave) radars, operate at frequencies in the range of 30 to 300 GHz. These radar systems are capable of providing fine spatial resolution and high sensitivity to small movements [42, 83]. This makes them particularly well-suited for sensing subtle micro-gestures. This research will leverage these properties of mmWave radars to explore the design, detection, recognition, and application of subtle mid-air gestures.

1.2 Thesis Statement

Accurate sensing of subtle mid-air micro-gestures using mmWave radar, and gesture recognition through signal processing and deep learning, enables quick, precise and user-friendly control of virtual interfaces, as demonstrated through empirical trials involving real-time user studies and statistical analysis of metrics such as recognition accuracy, task time, error distance and user experience scores.

1.3 Contributions

This thesis makes contributions in three key areas of radar-based gesture recognition and interaction. The first contribution focuses on the problem of addressing a system using subtle mid-air activation gestures. Frequency analysis techniques are used to examine how different hand motions produce distinct frequency signatures. Building on this, a novel activation gesture recognition pipeline is developed, utilizing spectral analysis to distinguish subtle activation gestures from incidental hand motions. Its effectiveness is evaluated through a user study assessing recognition performance and the ability to ignore unintended motions.

The second contribution focuses on the recognition of subtle micro-gestures using mmWave radar. In current radar-based gesture recognition research, there is a notable bias toward selecting macro-gestures. There is also a lack of publicly available radar gesture datasets that focus on subtle micro-gestures. In this work, a new dataset of subtle gestures is created, and the techniques for gesture detection and data collection are presented. Following this, deep learning is applied for gesture recognition, and various neural network architectures are trained and evaluated to recognize subtle gestures.

The true test of a gesture recognition system lies in its performance during live user interactions. While past research in radar-based gesture recognition has primarlity focused on developing gesture recogntion models, there is a lack of research involving real-time user studies. The final contribution moves from offline evaluations to practical, real-time assessments. This work integrates the neural network architectures developed in the previous contribution into real-time applications. A user study is then presented to evaluate the effectiveness of subtle gestures in controlling slider-based applications, providing quantitative and qualitative assessments of task performance and user experience.

1.4 Research Questions

This thesis aims to answer the following questions:

RQ1: What mid-air gestures are suitable as subtle activation gestures?

RQ2: How can subtle activation gestures be accurately recognized without extensive data acquisition?

RQ3: How accurately can neural networks recognize subtle gestures from mmWave radar data?

RQ4: Do subtle gestures improve task performance and user experience in radar-based interactions involving slider control?

1.5 Thesis Structure

Chapter 2, *Literature Review*, provides an overview of the historical context and recent advancements in mid-air gestural interaction and sensing technologies, particularly focusing on mmWave radars. It then reviews research involving mmWave radars in HCI, followed by the foundational principles of radar signal processing, which are essential for understanding the technical contributions in subsequent chapters. Finally, the chapter explores the different types of subtle interaction and the significance of subtle mid-air gestures in HCI.

Chapter 3, *Activation Gesture Recognition Using Frequency Analysis*, addresses **RQ1** and **RQ2**. It identifies suitable subtle activation gestures based on their ability to generate distinct high-frequency components, while remaining low-effort, visually unobtrusive, and spatially compact. A signal processing pipeline is introduced that can accurately detect these activation gestures without extensive data acquisition. A user study using multiple sensors (including mmWave radar) is presented, demonstrating that a frequency-based approach can recognize subtle activation gestures effectively while minimizing false activations.

Chapter 4, *Subtle Gesture Recognition Using Deep Learning*, addresses **RQ3**. It focuses on developing a new dataset of subtle micro-gestures and applying deep learning methods to recognize them from mmWave radar data. By training various neural network architectures, this chapter demonstrates the effectiveness of hybrid architectures in accurately recognizing different types of subtle gesture.

Chapter 5, *Exploring Slider Control Using Subtle Gestures*, addresses **RQ4**. Building on the findings of the previous chapter, it integrates the best-performing trained architecture into real-time interactive systems involving slider-based applications. A user study demonstrates that subtle gestures enable quick, precise, and comfortable control, improving task performance and usability compared to traditional macro-gestures.

Chapter 6, *Conclusion* revisits the thesis statement and the research questions, summarizing the key findings from each chapter. It discusses the contributions made to radar-based gesture recognition and interaction, acknowledges limitations, and outlines directions for future research.

2 Literature Review

2.1 Introduction

The first chapter briefly highlighted how recent advancements in radar sensing technologies have opened up possibilities for detecting subtle mid-air gestures. This literature review delves deeper into the radar technology employed in this research, examining how it enables precise, fine-grained motion sensing. In addition, this review will also explore what constitutes *subtle interaction*, examine the different types, and discuss where and why such interactions are beneficial in human-computer interaction (HCI).

In Section 2.2, the review will first provide historical context by exploring the development and appeal of mid-air gestures, highlighting how sensing technologies and mid-air gestural interactions have evolved over time. Following this, Section 2.3, will discuss the growing relevance of radars in HCI, and introduce the Google Soli radar, which is an important sensing technology in this research. Section 2.4 will then discuss radar fundamentals, outlining the principles behind signal generation, transmission, and processing. Radar data is complex, requiring specialized algorithms to interpret and extract meaningful information, particularly for tasks such as gesture recognition. Section 2.5 will focus on this aspect, delving into the machine learning algorithms used to process radar data for recognition of mid-air gestures. This collection of background research will provide the necessary technical foundation to understand the gesture recognition systems developed in the later chapters. Finally, Section 2.6, reviews the concept of subtle interaction. The discussion will include an exploration of the various types of subtle interaction, including

subtle mid-air gestures, and how they have been employed in previous research, analyzing its design principles, use cases, and benefits in different HCI scenarios.

2.2 Evolution of Mid-Air Gestural Interaction

A mid-air gesture is a type of touchless interaction where users control or interact with a smart device by performing specific hand movements in the air. These gestures involve deliberate, recognizable actions, such as swiping, pinching, waving, or pointing, and typically occur within a sensor-defined interaction zone. Importantly, mid-air gestures do not include incidental movements or non-communicative actions—such as scratching, stretching, or casual arm movements—that lack the explicit intent to interact with a device.

Mid-air gestures are particularly useful in scenarios where touch-based controls are impractical or unhygienic. For example, in medical settings, they support sterility, allowing hands-free control of equipment and displays [81]. In shared spaces like public kiosks, they can help minimize wear and contamination [82]. Additionally, mid-air gestures make it easy to handle tasks like adjusting TV settings or controlling music, allowing quick interaction without touching a device [96].

Gestural interaction is enabled through one of two types of sensors: *wearable* or *non-wearable* [54, 55, 4]. Wearable sensors require the user to physically wear a device to sense the gesture. These sensors are often capable of providing precise motion tracking and offer direct access to the user's movement. One of the first gesture-based systems to utilize wearable sensors was *Put-That-There* [9], developed in the late 1970s. It required users to wear a small sensor cube on their hand, which was tracked through an external magnetic field generated by a large transmitter. In the early 1990s, interaction using data gloves were introduced in *Charade* [7], a gesture-based system that interpreted hand movements to control presentation slides. These gloves provided a more direct way to track

fine-grained finger motions. The *SixthSense* project [49] in 2009 showcased a wearable gestural control system using a pocket projector and camera in a pendant-like device to project digital content onto surfaces. The system enabled users to manipulate projected digital content using gestures such as pinch-to-zoom and fingertip drawing.

Wearable sensors have seen success in both commercial applications and research. However, they introduce practical limitations, such as the need to be worn consistently and potential discomfort or intrusiveness. Non-wearable sensors provide an alternative approach to capturing mid-air gestures without requiring users to wear any additional devices. One early example of non-wearable gestural sensing was developed in the 1980s by Vincent John Vincent and Francis MacDougall.[1] They created a gesture-controlled musical system using computer vision to track full-body movements, enabling users to interact with virtual instruments through gestures. The *Microsoft Kinect*, released in 2010, was a major advancement in non-wearable gesture sensing. It used a color camera and depth sensor to track full-body movements in real-time. Initially designed for gaming, the Kinect's depth-sensing capabilities made it widely popular in research and commercial applications.

Up until the 2010s, non-wearable sensors primarily relied on optical cameras, often using visible or infrared light, which were effective for capturing *macro-gestures*. Macro-gestures are large, deliberate movements involving the whole arm or body, such as waving, swiping, or pointing. In the following years, non-optical alternatives based on radio frequency (RF) sensing began to emerge. Among these, early Wi-Fi-based systems detected motion by analyzing how human gestures disrupted wireless signal patterns. For instance, *WiSee* [60] and *WiGest* [1] demonstrated that fluctuations in Wi-Fi signals could be used to recognize gestures such as pushing, pulling, and swiping, even through walls. These systems operated over several meters and required no line-of-sight, leveraging existing wireless infrastructure to enable low-cost, always-on gesture recognition. These methods worked well for macro-gestures involving limb displacements on the order of tens of centimeters to over a meter, but lacked the spatial resolution needed to capture finer hand or finger

[1] `https://www.youtube.com/watch?v=-zQ-2kb5nvs`

movements.

As gesture interaction evolved beyond macro-gestures, there was growing interest in techniques that could detect more fine-grained, localized input. These smaller gestures—often referred to as *micro-gestures*—involve small movements of the hands and fingers within a compact interaction space, typically in the order of a few centimeters [92, 91, 10]. They are particularly useful in contexts where macro-gestures are socially awkward, fatiguing, or physically constrained. One of the first sensors to enable such fine-grained gesture sensing was the *Leap Motion Controller* from Ultraleap,[2] introduced in 2012. It uses two infrared cameras to reconstruct hand structure in 3D and detect depth and precise finger motions.

Acoustic sensing techniques also helped bridge the gap between macro and micro-input. Systems like *SoundWave* [26] and *FingerIO* [52] used inaudible ultrasonic signals emitted from device speakers, measuring the echoes received through microphones to detect motion. While *SoundWave* enabled directional gesture detection (e.g., hand swipes), *FingerIO* demonstrated sub-centimeter finger tracking by using sonar-like chirps and multiple microphones. Despite limitations in range and vulnerability to ambient sound, they marked a key step in sensing micro-gestures. Since the mid-2010s, low-cost, miniature radars have also emerged as a distinct class of non-optical sensors with very high spatial resolution and motion sensitivity to both macro and micro-gestures, expanding the possibilities for mid-air gestural interaction.

2.3 mmWave Radars in HCI

Radars have been a significant area of research and development since the 1940s, however, its use in HCI has only gained popularity in recent years. In the past, the requirement for specialized signal-processing techniques and custom hardware had created substantial barriers to entry. Recent advancements in affordable, miniaturized radar-on-chip technology and user-friendly SDKs have significantly lowered these barriers, with companies like Texas

[2] `https://leap2.ultraleap.com/products/leap-motion-controller-2/`

Instruments and Infineon Technologies developing low-cost commercial high-resolution millimeter-wave (mmWave) radars [61, 51, 30].

mmWave radars are a type of radar system that operate at millimeter-wave frequencies, specifically in the range of 30 to 300 GHz. They possess many properties that have made them increasingly popular in HCI research over the past few years. Unlike camera-based sensors, mmWave radars do not rely on visual information. They use radio waves, a type of electromagnetic signal, to detect the range and speed of objects. This allows them to sense objects in environments where visible light sensors may struggle, such as in low-light conditions, strong shadows, or varying ambient lighting. Additionally, they can sense through certain materials, such as clothing, plastic, or thin walls, enabling the detection of objects even when they are partially or fully occluded.

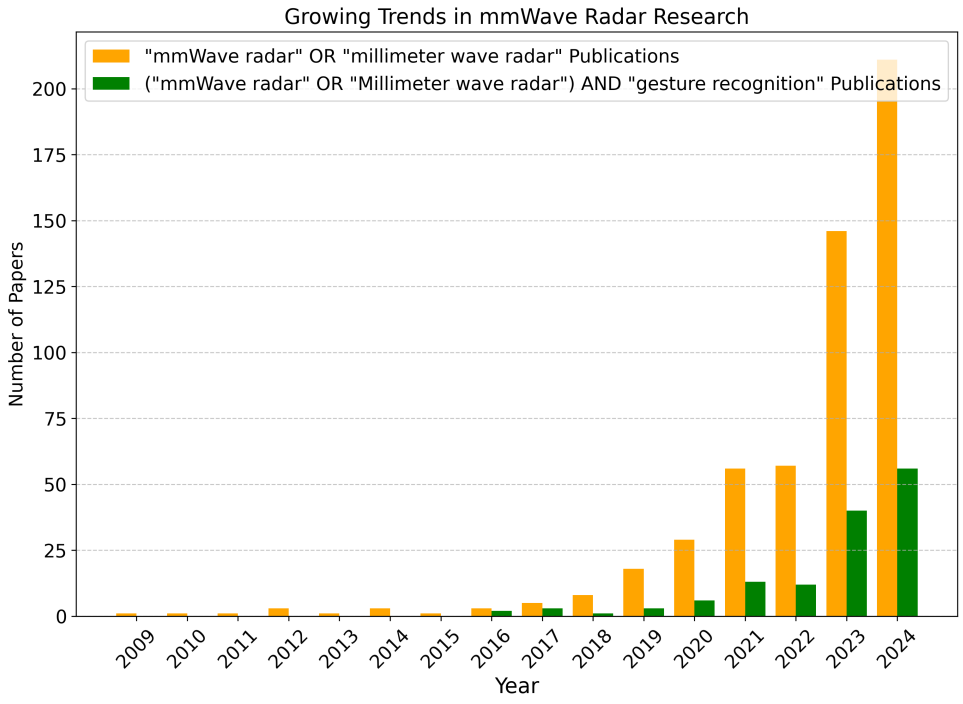


Figure 2.1: Number of publications in the ACM Digital Library over time matching the search queries: (1) “mmWave radar” OR “millimeter wave radar”, and (2) (“mmWave radar” OR “millimeter wave radar”) AND “gesture recognition”. The results reflect the growing interest in mmWave radar research, with a noticeable increase in gesture recognition applications from 2016 onward.

By adjusting the operating frequencies and configurations of mmWave radars, they can be optimized for detecting different types of motion at varying distances. For instance, at close

range (typically within 0–20 cm), mmWave radars can detect millimeter-level displacements, such as finger tremors. At longer distances (ranging from several meters to tens of meters, depending on the radar configuration), they can track larger body movements, such as walking, arm swings, or whole-body gestures. In long-range applications, mmWave radars have been used for human tracking, identification, and localization [68], as well as for gesture and activity recognition [2]. mmWave radars have also been applied in security contexts for detecting concealed weapons, utilizing their ability to penetrate clothing [50]. In short-range configurations, mmWave radars have been shown to be sensitive enough to capture human physiological signals, thereby enabling their use in health monitoring applications, such as heart rate and respiration measurement [70]. They have also been employed for material classification, where their sensitivity at close range allows for the accurate distinction between materials based on their unique radar reflections [94].

There is a growing trend in mmWave radar research, particularly in the area of gesture recognition. Figure 2.1 presents the number of publications in the ACM Digital Library between 2009 and 2024 that match two specific search queries: (1) “mmWave radar” OR “millimeter wave radar” (496 publications), and (2) (“mmWave radar” OR “millimeter wave radar”) AND “gesture recognition” (136 publications). The rapid rise in such publications—especially after 2016—reflects increasing research interest in this application area. The emphasis on gesture recognition in mmWave radar research has largely been due its capabilities in motion sensing. mmWave radars offer fine spatial resolution and sensitivity, enabling the detection of both macro- and micro-gestures. Another key factor driving this focus is the privacy-preserving nature of radars. Since they do not record any visual data, mmWave radars address many privacy concerns associated with optical systems, which capture visual details of users and their surroundings. In contrast, mmWave radars infer only *range* and *velocity* information without recording any visual data, making them much more *privacy-preserving*. Additionally, the miniaturization of mmWave radar technology has played a pivotal role in its adoption for gesture recognition. The development of small, low-power radar chips like the *Google Soli* has opened up the

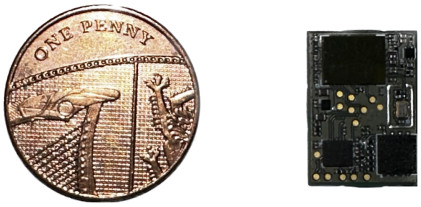
possibility to integrate radars into a wide range of devices.

2.4 Understanding mmWave Radar Fundamentals Using Google Soli

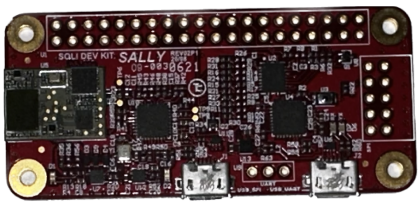
Soli is a high-resolution, low-power, miniature mmWave radar, designed by Google's Advanced Technology and Projects (ATAP) group [42]. It utilizes the Infineon BGT60TR24 frequency-modulated continuous-wave (FMCW) radar chip, which operates at a center frequency of 60 GHz, placing it in the millimeter-wave spectrum. The chip has been modified by Google ATAP to optimize for low power consumption and enhanced signal processing capabilities, allowing it to take rapid measurements of movement—up to thousands of times per second—that enable precise detection and tracking of fine-grained movements. To achieve low power consumption, adaptive duty cycling was implemented, which involves shutting down the chip during computation phases to save energy. For enhanced signal processing, the design includes high-speed analog-to-digital converters (ADCs) and custom digital signal processing (DSP) algorithms. Specifically, an Infineon XMC4500 Cortex M4 microprocessor with quad 12-bit ADCs running at 1.79 Msample/sec is used, enabling the radar to process signals at higher frame rates necessary for tracking objects with *sub-millimeter accuracy*. Figure 2.2a shows the Soli chip, with the integrated DSP and ADC chips. Figure 2.2b illustrates the Soli development kit, where the Soli chip is mounted on an extended breakout board.

Soli possesses all the properties of mmWave radar discussed in the previous section, such as the ability to sense through materials and operate in various lighting conditions. What sets it apart from other mmWave radars is its low power consumption and compact size, which has enabled its integration into commercial consumer devices, such as the Google Pixel 4[3] smartphone and the Google Nest Hub.[4] In the Google Pixel 4, Soli enabled

[3] `https://blog.research.google/2020/03/soli-radar-based-perception-and.html`
[4] `https://support.google.com/googlenest/answer/10388741?hl=en-GB`



(a) The Soli radar chip (compared to a penny for size reference).



(b) The Soli dev kit comprising of a Soli chip mounted on an extended breakout board, designed to facilitate connections and development with external devices, such as laptops.

Figure 2.2: The Soli radar system.

mid-air interactions using macro-gestures such as hand swipes to skip songs, snooze alarms, and silence phone calls. In the Google Nest Hub, Soli was used infer when a user is paying attention to the device through head tracking. It also enabled sleep tracking abilities that could detect breathing patterns and assess quality of sleep.

Another important property of Soli is its fine *range resolution*. Range resolution refers to the minimum distance between two objects that a radar can differentiate. It is determined by its *bandwidth*. According to the equation for range resolution:

$$res_r = \frac{c}{2BW}, \tag{2.1}$$

where c is the speed of light and BW is the bandwidth of the radar [73]. The wider the bandwidth, the finer the radar's ability to distinguish between closely spaced objects. For Soli, the maximum permitted bandwidth is **7 GHz**, which gives it a range resolution of approximately:

$$res_r = \frac{3 \times 10^8 \,\mathrm{m/s}}{2 \times 7 \times 10^9 \,\mathrm{Hz}} = 0.0214 \,\mathrm{m} = 2.14 \,\mathrm{cm} \tag{2.2}$$

This means that Soli can differentiate between two objects that are as close as 2.14cm apart. There is no strict minimum or maximum size of object that Soli can detect; rather,

detectability depends more on the object's reflectivity and how effectively it reflects radar signals—commonly referred to as its radar cross-section. Objects that reflect more energy back to the radar are easier to detect, regardless of their physical size. In practice, Soli is capable of sensing small, dynamic features such as fingertips, especially when they are in motion.

Figure 2.3 gives an overview of how Soli operates, detailing the processes of signal generation, transmission, and the transformation of received signals into meaningful information. The following sections will elaborate on each of these stages in detail. While this explanation is specific to the Soli, the principles and techniques discussed are also relevant to other mmWave radars.

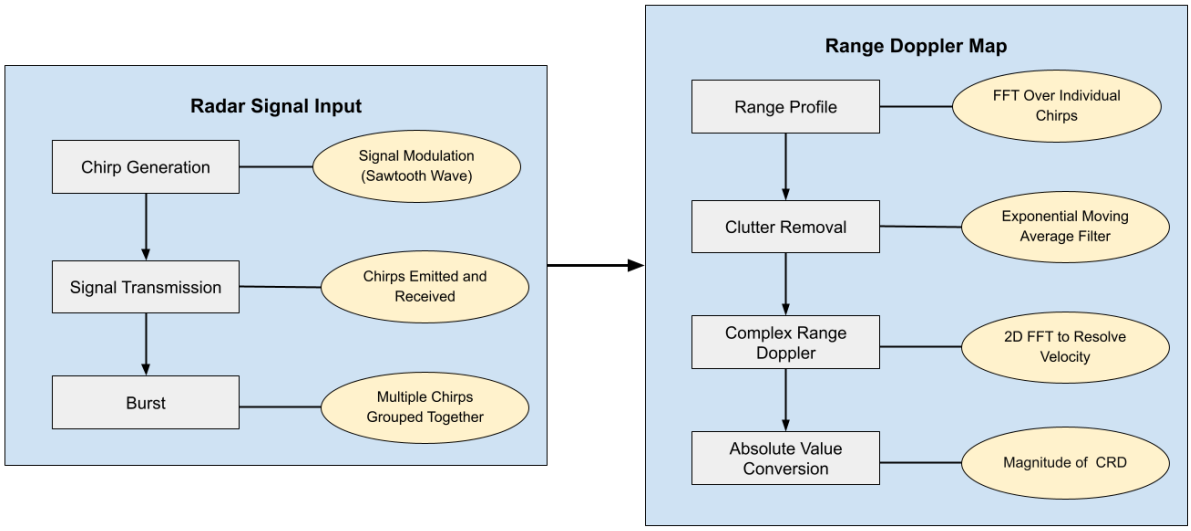


Figure 2.3: Signal generation and processing pipeline of a mmWave radar system. The left section illustrates the generation, transmission, and grouping of radar chirps, while the right section outlines the process of resolving range profiles, removing clutter, and applying FFTs to extract range and velocity information, ultimately forming a range-Doppler map.

2.4.1 Signal Generation and Transmission

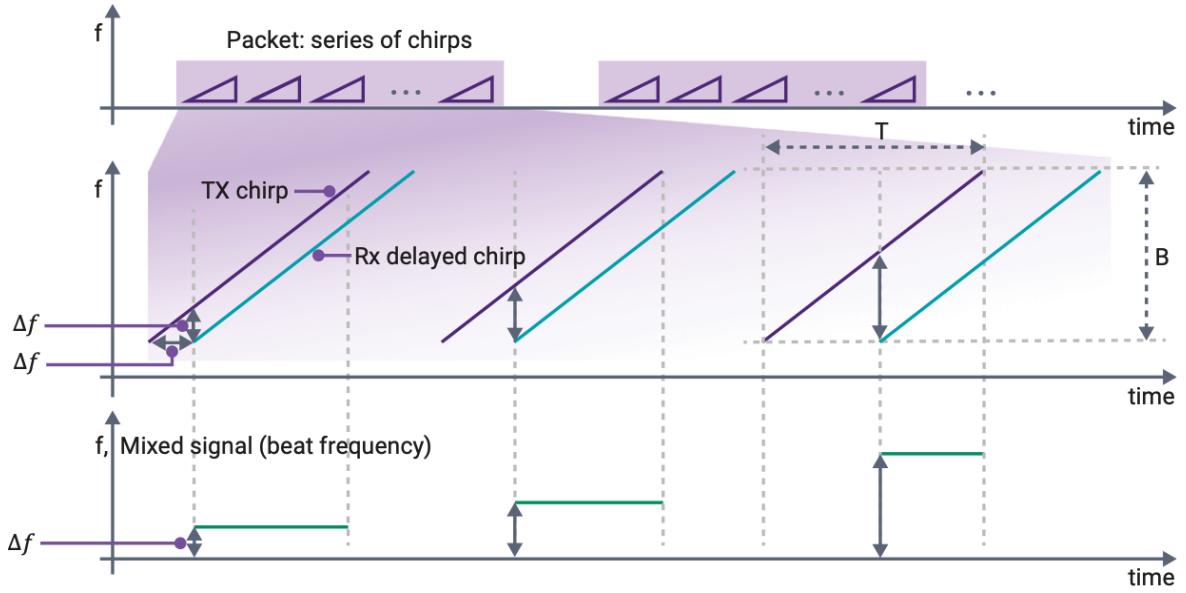
The Soli features an Infineon 60GHz FMCW radar with one transmission antenna and three receiver antennas. FMCW radar functions by emitting a modulated electromagnetic wave toward a target, which then scatters the transmitted signal, with some portion of

energy redirected back toward the radar. Waves are emitted by transmitting a series of *chirps* from the transmission antenna. A chirp in the Soli follows a sawtooth pattern (see Figure 2.4a). This means that each chirp starts at a base frequency and linearly sweeps up to a higher frequency over a short period of time, then quickly resets to the base frequency to begin the next sweep. The Soli transmits and receives a series of chirps which are grouped into what are called packets or *bursts*. By analyzing the time delay and frequency shift of the received chirps, the distance, and velocity characteristics of the target can be resolved [77].

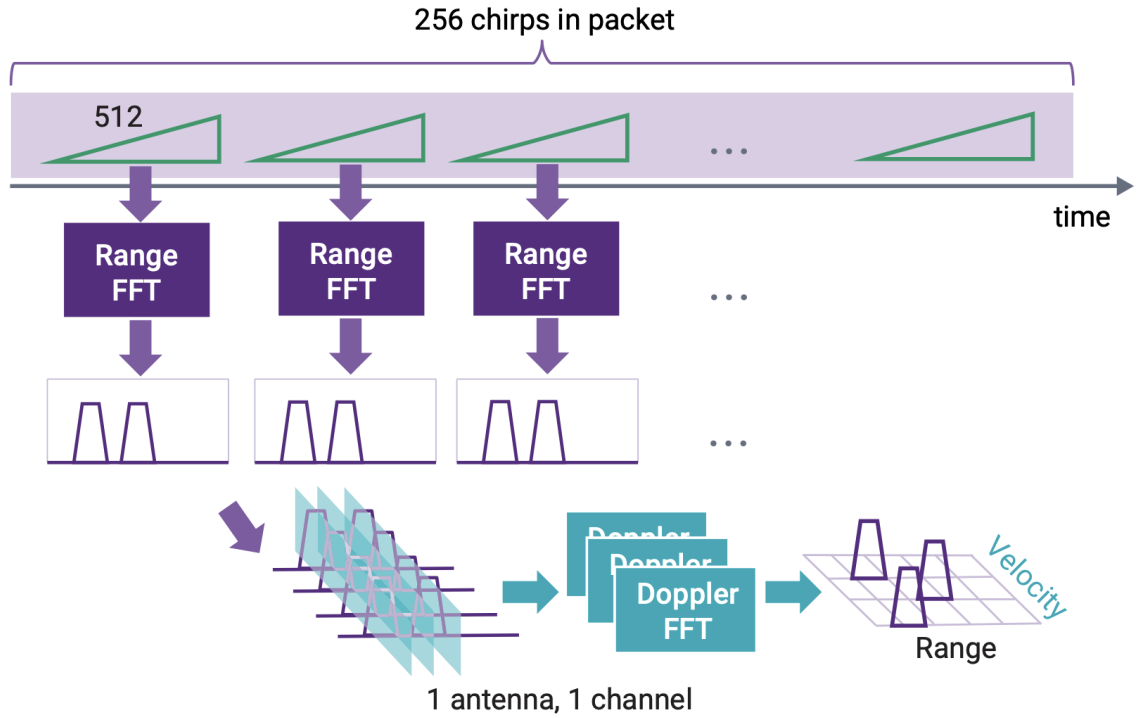
2.4.2 Resolving Range

Time Delay and Beat Frequency: When a chirp is reflected from an object, it travels back to the radar with a time delay. This time delay causes a difference in frequency between the transmitted chirp at the time of transmission and the received chirp at the time of arrival. This difference in frequency is known as the *beat frequency*, which is directly proportional to the distance of the object because the greater the distance, the longer the time delay, and thus a larger frequency difference [77].

Applying FFT to Chirps: The radar system processes the received chirp using a *Fast Fourier Transform* (FFT). The FFT translates the time-domain signal (which varies in frequency over time) into a frequency-domain signal, representing the intensity of different frequencies. The peak(s) in the FFT spectrum represent the beat frequency (see Figure 2.4b). Since the beat frequency is related to the range, detecting this peak allows the calculation of the object's distance from the radar. The result of this processing is known as the *range profile*. The range profile is a data representation that shows the intensity of reflected signals as a function of distance from the radar. It provides a detailed view of the distance distribution of objects in the radar's field of view. Each peak in the range profile corresponds to a detected object, and the position of the peak indicates the radial distance (straight-line distance from the radar) to the object [73].



(a) Transmission of a series of chirps and the resulting beat frequency in FMCW radar system. The TX chirp represents the transmitted signal, and the RX delayed chirp represents the received signal reflected from a target. The difference in frequency (Δf) between the TX and RX chirps forms the beat frequency, which is used to calculate the range and velocity of the target.



(b) A visual overview of signal processing in an FMCW radar system. A packet (or burst) of 256 chirps is transmitted, and each chirp undergoes Range FFT to extract the range information from the beat frequency. Subsequently, a second FFT, referred to as Doppler FFT, is applied across the chirps in the packet to resolve velocity, resulting in a range-Doppler map.

Figure 2.4: Signal generation and processing in an FMCW radar system. (a) shows how the beat frequency is generated from transmitted and received chirps, while (b) depicts the process of applying FFT to extract range and velocity information. Images taken from [77].

2.4.3 Clutter Removal

Clutter refers to unwanted echoes from stationary or slow-moving objects within the radar's field of view, which can obscure or interfere with the detection of relevant targets [73]. In the signal processing for Soli, clutter removal is implemented by calculating an exponential moving average of the received range profiles across multiple chirps. This averaged signal, or *clutter map*, is then subtracted from each individual chirp's data in order to suppress the clutter and improve detection of the true targets.

2.4.4 Resolving Velocity

Doppler Shift: If the object is moving, the frequency of the received echo will also be shifted. This is known as the *Doppler effect*. This shift is added to or subtracted from the beat frequency, depending on whether the object is moving toward or away from the radar. Positive Doppler shift indicates motion away from the radar (increasing distance), and negative shift indicates motion toward the radar (decreasing distance).

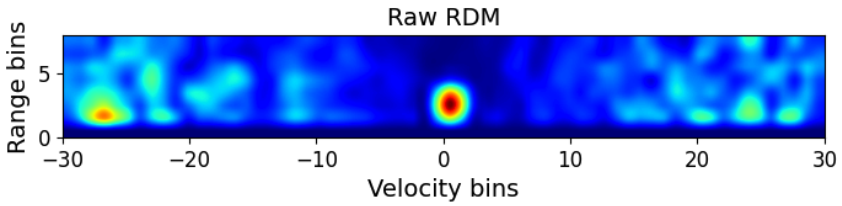
Applying 2D FFT to Bursts: To separate the range information from the velocity information, a two-dimensional FFT can be applied. The first FFT is applied along the samples within a single chirp to extract the range profile, as mentioned in Section 2.4.2. The second FFT is applied across the series of chirps in a burst. By transforming the changes in beat frequencies over successive chirps into the frequency domain, the FFT reveals how these frequencies vary over time, indicative of the object's velocity [73].

2.4.5 Range-Doppler Map

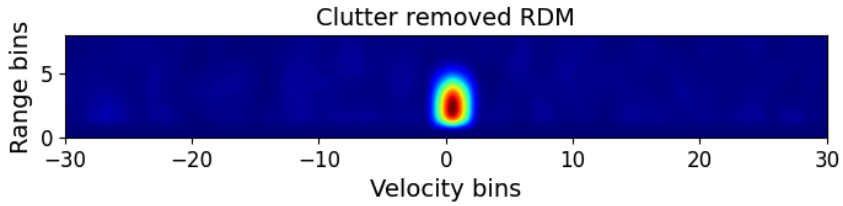
The direct output of the 2D FFT is a complex range-Doppler map. The complex values contain both the amplitude (magnitude) and phase information of the radar signal returns at different ranges and Doppler shifts. To create a visual representation that is interpretable, only the magnitude of the complex numbers is taken and can be visualized as a *range-*

Doppler map (RDM). A RDM exhibits reflected energy intensity as a function of target range and velocity. Positive velocity corresponds to motion away from the radar (increasing range) and negative velocity corresponds to motion toward the radar (decreasing range) [77, 48].

Figure 2.5 shows examples of RDMs before and after clutter removal. Figure 2.5a displays the raw RDM, where the presence of clutter obscure the signal of the actual target. Figure 2.5b shows the RDM after applying clutter removal as described in Section 2.4.3. The clutter removal process filters out unwanted reflections, enhancing the clarity of the target signal and allowing for a more accurate representation of the range and velocity of the actual target.



(a) Raw RDM showing the intensity of reflected signals as a function of range and velocity bins before clutter removal. The presence of noise and stationary objects obscures the target signal.



(b) Clutter-removed RDM after applying filter to eliminate unwanted reflections. The target signal is more prominent, showing a clearer distinction of the reflected energy.

Figure 2.5: Comparison of RDMs before and after clutter removal.

2.5 Gesture Classification From Radar Data

Classification is a fundamental task in machine learning where the goal is to assign input data to one of several predefined categories. This process involves training a model on a labeled dataset so that it can learn to differentiate between various classes based on the

features of the input data.

The general classification function can be expressed as follows:

$$\hat{y} = \text{argmax}_c \, f(x; \theta),$$

where:

- $\hat{y}$ is the predicted class label.
- c represents the possible classes.
- x is the input data.
- $f(x; \theta)$ is the classification model parameterized by θ.
- The argmax function selects the class c with the highest predicted probability.

In deep learning, the function f is modeled by a *neural network*. Neural networks consist of multiple layers of interconnected neurons, each layer transforming the input data into a higher-level representation. During training, the model is optimized by adjusting θ to minimize a loss function, to improve its ability to predict the correct class for new, unseen data. The parameters θ include the weights and biases of all the neurons in the network.

2.5.1 Neural Networks for Gesture Classification

Neural networks have been effectively employed in various domains, such as image understanding [63, 37], speech recognition [98], machine translation [78], and localization [21]. Neural networks have also been widely used in gesture classification, particularly with mmWave radars. The input to a network in this context is typically a RDM sequence, which provide information on the reflected energy intensity across different ranges and velocities over consecutive time frames. As a gesture is performed, the RDM changes dynamically, reflecting the movement of different parts of the hand or body. Neural networks are then able to learn the temporal and spatial patterns within these RDM sequences, and classify different gestures.

Table 2.1 provides a summary of the different radar sensors, the types (macro/micro) and numbers of gestures recognized, and the neural network architectures employed from various studies. The classification of gestures into macro and micro categories was based on the movement scale observed in each study. As mentioned in Section 2.2, macro-gestures refer to large, deliberate movements involving the whole arm or body—typically with displacements on the order of tens of centimeters or more—whereas micro-gestures involve smaller, localized hand and finger movements, typically within a compact interaction space in the order of a few centimeters. Recognizing these gestures requires models capable of capturing both spatial and temporal features. Two classic deep neural network models are particularly popular for gesture recognition from mmWave radar data: the *convolutional neural network* (CNN) and the *long short-term memory network* (LSTM).

- CNNs are specialized neural networks designed for spatial data processing, particularly well-suited for tasks like image recognition. They use convolutional layers to detect spatial patterns, such as edges, shapes, and textures, making them effective for extracting features from RDM data.
- LSTMs are a type of recurrent neural network (RNN) designed to process sequential data by maintaining memory over time. Unlike standard RNNs, LSTMs can capture long-term dependencies in temporal data, making them ideal for tracking motion patterns in radar data sequences.

Dong et al. use a 3D-CNN architecture to recognize 16 macro-gestures. 3D-CNN is an extension of traditional CNNs into the time dimension, allowing it to capture both spatial and temporal features. Choi et al. use an LSTM encoder to learn the temporal characteristics of the RDM sequences and recognize 10 macro-gestures. In addition, there are many studies [83, 29, 86] that have applied hybrid CNN-LSTM networks, which benefit from the strengths of both CNN and LSTM, with CNNs effectively capturing spatial features (such as shape and movement direction) and feeds these to the LSTM which captures temporal dependencies (such as speed and rhythm). These hybrid models tend to have higher recognition accuracy than using CNN or LSTM alone. More complex

networks such as *VGG-Net* [85], *ResNet* [47], and *Transformer* [13] have also been adapted for mmWave gesture recognition. These architectures bring the feature extraction and sequence modeling capabilities used in other tasks and transfer them to recognizing gestures from radar data.

Radar	**Gestures/Number**	**Classification Algorithm**	**Reference**
IWR1642	Macro/9, Micro/1	3D-CNN	[69]
AWR1642	Macro/16	3D-CNN	[19]
Soli	Macro/10	LSTM	[14]
IWR1443	Macro/6	LSTM	[93]
Soli	Macro/7, Micro/4	CNN-LSTM	[83]
Soli	Macro/4	CNN-LSTM	[29]
AWR1642	Macro/8	CNN-LSTM	[86]
AWR1642	Macro/6	VGG-16	[85]
AWR1843	Macro/3, Micro/3	2D + 3D ResNet18	[47]
Soli	Macro/20	Transformer	[13]

Table 2.1: Radar sensors, gesture types, and neural networks used in gesture recognition studies.

Although deep learning models have demonstrated strong performance in radar-based gesture recognition, they have been predominantly applied to large, well-defined macro-gestures. Several studies have reported classification accuracies exceeding 95% [19, 47, 14]. These results are often aided by the more distinct motion signatures of macro-gestures, which involve larger arm or hand movements and produce stronger, more separable radar signals. This thesis shifts the focus toward micro-gestures, evaluating whether these existing models remain effective when applied to smaller, localized hand and finger movements. In doing so, it also explores the design of micro-gestures that are both distinguishable to the radar and usable in practical contexts. Furthermore, unlike prior work which primarily evaluates recognition accuracy in offline settings, this research also explores real-time implementation and usability by integrating the models into live applications and conducting user studies to assess their practical viability.

2.5.2 Radar Gesture Datasets

Deep learning models require large amounts of labeled data to effectively learn and generalize from the underlying patterns in the input. This is because neural networks have millions (even billions) of parameters that need to be optimized during training. Without sufficient data, these models fail to generalize to new, unseen inputs. Currently, there are a limited number of publicly available radar gesture datasets, particularly those that focus on subtle micro-gestures. Table 2.2 shows a summary of the available datasets. These datasets tend to focus on macro-gestures such as swipes, waves, and rotations. Similarly, as was seen in Table 2.1, research in gesture recognition models also tend to favor macro-gestures.

The focus on macro-gesture recognition has been driven by several reasons. These gestures produce clearer and more distinct radar signatures, making them easier to detect and train neural networks on, resulting in higher recognition accuracy. Additionally, since these gestures are performed with larger body movements, such as the entire arm, they generates strong radar signals that can be detected from several meters away. Because of this, macro-gestures can be recognized with high accuracies from long ranges. For example, Liu et al. demonstrated a 95% gesture recognition accuracy for four macro-gestures performed from a distance of 2.4 meters.

Radar	**Gestures/Number**	**Total Samples**	**Reference**
Novelda XeThru X4	Macro-gestures/11	9600	[3]
Ancortek	Macro-gestures/2, Micro-gestures/2	3052	[64]
IWR1443 BOOST	Macro-gestures/2, Micro-gestures/3	56,420	[44]
Soli	Macro-gestures/7, Micro-gestures/4	27,500	[83]

Table 2.2: Publicly available datasets for mmWave hand gesture recognition

2.6 Understanding “Subtle”

In the Introduction of this thesis, the concept of subtle gestures was briefly introduced. But what exactly does “subtle” mean in this context? This section aims to expand on this by discussing subtle interaction and what constitutes subtle gestures within the wider concept of subtle interaction, as well as the reasons behind their importance in HCI.

2.6.1 Types of Subtle Interactions

Pohl et al., in their paper “Charting Subtle Interaction in the HCI Literature”, analyzed 55 HCI publications that used the term “subtle”. They reviewed these publications to identify common themes and categorize subtle interaction. Based on this analysis, four main types of subtle interactions were found and each type reflects different qualities and design goals. These are:

1. **Non-Intrusive Feedback:** Non-intrusive feedback in the context of subtle interaction means not drawing much attention away from whatever the user is doing. The goal is to provide subtle cues that inform or notify the user without significantly disrupting their primary activity or requiring them to shift their focus entirely. Hansson and Ljungstrand explored this with their *Reminder Bracelet*, which uses light, color, and patterns instead of sound to notify users [28]. Similarly, Costanza et al. developed an eyeglass peripheral display that delivers visual cues in the wearer’s periphery which notify users of incoming messages, reminders, or alerts through subtle changes in light patterns and colors, allowing the user to stay informed without significantly disrupting their focus [16].
2. **Low-Effort Input:** Subtle interaction can also mean doing less, emphasizing the reduction of physical effort required for input. Such interactions are tied to small-scale or low-amplitude gestures, performed with minimal exertion, also commonly referred to as *micro-gestures*. For instance, Costanza et al. described subtle input as requiring “very little or no movement at all” [17]. The *Gunslinger* system illustrates this by

proposing small, arms-down mid-air gestures, such as thumb and finger movements, aiming to minimize physical input space and reduce user fatigue [45]. Similarly, the *WristFlex* system uses minor hand movements, such as pinching two fingers [18]. The *Nenya* ring is another example which enables control through “small, discreet movements” [6]. The EMG controller by Costanza et al. detects subtle gestures from muscle contractions using surface electromyography, recognizing small hand movements through muscle activity signals [17]. The *FingerPad* by Chan et al. facilitates low-effort input with small touch gestures on a hidden touchpad embedded under surfaces like tables or clothing [11].

3. **Discreet Interaction:** This form of subtle interaction focuses on hiding actions from others. This can be particularly important in social contexts where overt interaction with devices might be considered rude or distracting. Devices like the FingerPad enable users to interact discreetly by using a touchpad embedded under the surface of a table or a piece of clothing, allowing for private interactions that are hidden from view [11]. Similarly, the EMG controller leverages the discreet nature of muscle contractions, detecting subtle movements on the bicep to issue commands [17]. Discreet interaction is closely tied to low-effort input, as both emphasize small-scale or low-amplitude movements that are minimally visible and require little physical exertion. Systems like FingerPad and EMG controller, highlight this overlap, as the low-effort nature of these interactions inherently supports discreet usage.
4. **Nudging Users:** Finally, subtle interactions can be used to nudge users towards certain behaviors or actions. This approach leverages cues to guide users’ attention and actions. For example, a subtle nudging technique was implemented by Sridharan et al., where brief and localized adjustments in screen brightness or contrast were used to influence where a user looked without the user being fully aware of the manipulation [72].

2.6.2 Subtle Gestures

From the broader taxonomy of subtle interaction, *subtle gestures* can take on multiple interpretations depending on the context in which they are performed. For example, subtle interaction includes the concept of *discreet interaction*, where actions are hidden from others. In this sense, even a broad macro-gesture can be subtle if performed in a socially inconspicuous manner. One example of this is Google's *Project Jacquard* [58], which integrates capacitive touch sensors into clothing to enable gesture-based input on textiles. In their Levi's jacket collaboration [5], users could swipe the cuff of their sleeve to control music playback or receive navigation prompts. To an outside observer, the gesture may simply look like someone adjusting their sleeve. In such cases, subtlety arises not from the gesture's physical characteristics but from its social invisibility and natural integration into everyday movement.

However, in HCI literature, the term "subtle gesture" is more commonly associated with low-amplitude movements, a quality closely aligned with the concept of *micro-gestures*. Chan et al. [10] defined micro-gestures as "detailed gestures in a small interaction space," emphasizing miniaturization of hand movements for discreet interaction. In contrast, Wolf et al. framed micro-gestures as small hand and finger movements that can be carried out concurrently with another task [92, 91]. Their work included scenarios such as gesturing while holding a steering wheel, showcasing how such input could be performed without disrupting the primary activity.

Drawing from these perspectives, this thesis uses the term *subtle gesture* to refer primarily to low-amplitude hand and finger motions that are spatially compact, minimally demanding in terms of physical exertion, and socially unobtrusive. While the perception of effort is inherently relative and may vary across individuals or contexts—for instance, what feels effortless for one user may be challenging for another—this work adopts the term "low-effort" to describe gestures that, for most users, require minimal muscular activation

[5] `http://global.levi.com/jacquard/jacquard-with-buy-link.html`

and can be performed repeatedly without significant fatigue. Subtlety in this context arises from characteristics such as compactness, precision, and unobtrusiveness, making such gestures viable for frequent use in mid-air interaction.

2.6.3 Significance of Subtle Mid-Air Gestures

Subtle mid-air gestures are linked to low-effort input, emphasizing the reduction of physical effort required for interaction. These gestures use small, low-amplitude movements that are primarily driven by the muscles controlling the fingers and wrist. In Preethi's scenario (from Section 1.1.1), she experiences fatigue from performing large, repetitive macro-gestures to control the TV. By contrast, subtle gestures require far less exertion, making them more sustainable for frequent interactions.

The low-effort nature of subtle gestures also aligns with the concept of balancing between focused and casual interactions, as outlined by Pohl and Murray-Smith [57]. The focused–casual continuum reflects how users shift their level of engagement depending on context. For instance, a user cooking in the kitchen may casually wave a hand to skip a song. However, if the same user wants to rewind or scrub to a precise point in a podcast, this would require more focused attention and finer motor control. In such cases, subtle gestures offer a low-effort mechanism for precise input without requiring the user to break their primary activity entirely. By supporting both casual and focused modes of interaction, subtle gestures can enable more fluid transitions between task engagement and control.

Subtle mid-air gestures are also linked with the idea of discreet interaction to improve social acceptability. These gestures are easier to perform in public settings without disturbing others or breaking social norms. The work by Williamson discusses the concept of everyday actions as performances, where users are constantly aware of their surroundings and adjust their behavior based on social feedback. In Preethi's scenario, she was self-conscious of using large gestures in the presence of other people, as it could draw unwanted attention

and potentially cause embarrassment. Subtle gestures, in such scenarios, would reduce the visibility of interaction, which makes users more likely to engage in gestural interaction around other people [89].

2.6.4 Sensing Subtle Mid-Air Gestures

In the past, sensing subtle gestures heavily relied on wearable sensors, which provided high accuracy for detecting small movements but required users to wear additional devices. The Gunslinger system required a Leap Motion Controller to be mounted on the user's thigh to track small, arms-down mid-air gestures [45]; the WristFlex system used an array of force-sensitive resistors worn around the wrist to detect subtle finger pinch gestures by sensing tendon movements at the wrist [18]; the EMG controller used surface electromyography to detect muscle activity from electrodes placed around the upper arm, specifically centered on the bicep brachii [17]; the FingerPad device uses a nail-mounted touch sensor [11]. Several ring-based systems have also been developed for subtle gesture recognition [6, 33, 75, 76].

Early non-wearable sensors lacked the spatial resolution needed to detect fine, low-amplitude movements. Prior to the development of mmWave radars, Wi-Fi [95, 1, 31, 80] and ultrasound [84, 65, 12, 43, 26] signals were widely used for gesture sensing. Both approaches enabled non-wearable mid-air gesture sensing without requiring specialized hardware. Wi-Fi leverages commodity routers, while ultrasound utilizes built-in speakers and microphones, applying the Doppler effect—similar to mmWave radar—to detect motion. These methods performed well for detecting large hand motions and human posture, but they lacked fine spatial resolution.

With the development of low-cost, miniature mmWave radars like Google Soli, sensing subtle gestures has become more feasible. These sensors offer sub-millimeter accuracy and fine-grained range resolution, enabling the detection of even the smallest finger movements. Their sensitivity has been demonstrated in applications such as non-contact vital sign

monitoring [70] and distinguishing between different textures and materials [94]. The combination of high accuracy, fine range resolution, and compact form factor opens up new possibilities in subtle gesture sensing.

2.7 Conclusion

This literature review began by motivating the use of mid-air gestures in HCI. It then introduced mmWave radars and its increasing significance in HCI research. Following this, the review detailed the fundamental principles of radar signal processing and data representations that will be required to understand the technical contributions discussed in the upcoming chapters. A review of radar-based gesture recognition was then presented, revealing a bias in current research toward macro-gestures. This bias was evident not only in the recognition models but also in publicly available datasets, which tend to focus on macro-gestures. These gestures produce strong radar signatures, making them straightforward to detect and train on, thereby enabling high recognition accuracy but leaving subtle micro-gesture recognition largely underexplored.

The review then shifted focus to subtle interaction, outlining different forms such as non-intrusive feedback, discreet input, and user nudging. Within this space, subtle gestures were defined as small, localized hand or finger movements that are less physically demanding and more socially appropriate in shared or constrained environments. While wearable sensors have traditionally been used to detect such gestures, recent advances in high-resolution, low-power radars—such as Google Soli—offer new possibilities for sensing these gestures without requiring users to wear additional hardware.

The review identified key gaps in the mmWave radar literature: a lack of focus on subtle mid-air gestures, including limited evaluation of gesture recognition models on subtle gestures, and an absence of user-centered studies assessing real-time usability. These gaps are important because subtle gestures offer advantages in comfort, repeatability, and social

acceptability, particularly in everyday settings where macro-gestures may be awkward or tiring. Evaluating subtle gesture systems in real-time is also essential for understanding practical challenges like unintended activations, latency, and usability.

With the combination of sub-millimeter accuracy, fine range resolution, and small form factor, mmWave radars like Soli have great potential for enabling new forms of subtle interactions. The remainder of this thesis addresses the key gaps by designing subtle mid-air gestures tailored for radar sensing, developing real-time recognition systems, and evaluating their effectiveness and usability through controlled user studies.

3 | Activation Gesture Recognition Using Frequency Analysis

3.1 Introduction

Activation is the process of signaling the start of an interaction, ensuring that a system responds only when intended. In sensor-based interaction, activation serves as a crucial first step, determining when a system transitions from passive sensing to active engagement. However, this seemingly simple step often introduces unexpected challenges. Consider Preethi's scenario introduced in the beginning of this thesis (Section 1.1.1). Her gesture was unintentionally recognised and activated other devices in the room. Sensors are "always on" and as a consequence, every user motion may be interpreted as an interaction, whether or not it was intended. This is know as the *Midas Touch* problem in gestural interaction. Baudel and Beaudouin-Lafon also refer to this as *immersion syndrome* where the user is immersed in interaction, even if they do not want to be. The technical term for this is *'false positive errors'*, where the user does not intend to perform a gesture but the system recognizes one anyway.

Bellotti et al. identified the importance of being able to address a sensing interface, so that the system is able to discern when it should pay attention to a user's actions, and when to ignore them. One strategy for addressing the system is through the use of *activation zones*, where gestures are only recognized when performed within specific spatial regions near a screen or sensor [7, 24, 71]. This spatial filtering helps to disambiguate intentional gestures from casual movements occurring elsewhere. More generally, gesture recognition systems

often use *delimiters* that define the start and end of interaction segments, ensuring the system recognizes only the intended gestures within these boundaries [36]. An *activation gesture* (or *gating gesture* [90]) is a specific kind of gesture delimiter that is used to initiate the gesture recognition mode for further interaction. These gestures serve as a deliberate signal from the user to the system, indicating the start of intentional interaction. An activation gesture needs to be distinctive from everyday movements and unlikely to be performed accidentally [24, 35]. They function similarly to verbal hotwords used in voice-activated systems, such as "Hey Siri" or "OK Google," but in the form of physical movements.

Depending on the sensor type, activation gestures can be dynamic movements or static poses. Vision-based sensors often use static hand or body poses for activation since they benefit from the ability to reconstruct spatial structure [66, 82]. With motion sensors, like mmWave radars, it is not possible to directly reconstruct spatial structure and therefore the activation gestures must rely on dynamic movements [32, 34, 40]. Some gesture-based systems use *temporal correlation gestures*, where a user's movement is matched in time to a dynamic system cue, such as tracking a moving target on a screen [79]. These techniques can serve as an implicit form of activation, relying on synchronous motion as a signal of intent rather than requiring a distinct gesture.

Neural networks have demonstrated strong performance in recognizing macro-gestures like hand waves, fast swipes, and circular motions from mmWave radar data [83, 14, 29]. While these gestures are replicable and distinguishable, they are large motions that can lead to Midas Touch, as seen in Preethi's scenario when other sensing devices are present. Additionally, these gestures are neither low-effort when performed frequently nor discreet. The first research question in this chapter aims to identify subtle, dynamic activation gestures for sensors like mmWave radars.

RQ1: What mid-air gestures are suitable as subtle activation gestures?

After identifying appropriate activation gestures, the next challenge lies in designing algorithms that are sensitive enough to recognize these gestures and ignore incidental movements. As reviewed in Section 2.5, gesture recognition research with mmWave radars has primarily used deep learning algorithms, which rely on large amounts of data. This method is useful for gesture sets because neural networks can learn to differentiate between each gesture given enough training data. However, for activation gestures, the goal is to initiate interaction with the system, so not every detected motion segment needs to be processed by a neural network to identify whether it matches a gesture from an entire set. This would be highly inefficient, and especially problematic in battery-powered devices like smartphones, where battery life is limited. If the phone was to run every detected motion segment through a neural network, it would unnecessarily increase power consumption. Moreover, activation gestures are inherently time-sensitive, often requiring immediate system response. Offloading gesture recognition to cloud services introduces latency and depends on stable internet connectivity, which cannot always be guaranteed—especially in mobile or offline scenarios. Therefore, local, lightweight processing is preferable for activation gestures to ensure real-time responsiveness and energy efficiency. By focusing on these unique constraints, this chapter will develop a system for recognizing subtle activation gestures without the need for extensive data collection or calibration (typically required for neural networks) through the following research question:

> **RQ2:** How can subtle activation gestures be accurately recognized without extensive data acquisition?

3.1.1 Chapter Structure

Section 3.2 introduces the frequency analysis of hand motions, explaining how different types of hand movements generate distinct frequency components. Section 3.3 describes the selection of subtle activation gestures, followed by Section 3.4, which outlines the development of an *activation gesture recognition pipeline* that utilizes frequency analysis

to recognize subtle activation gestures without data acquisition. Section 3.5 presents the evaluation of the proposed pipeline through a user study. Section 3.6 summarizes the main findings of the evaluation, highlights the limitations and suggests areas for future work. Finally, Section 3.7 summarizes the chapter by revisiting the research questions and outlining the contributions made.

3.2 Frequency Analysis of Hand Motions

Our hands are capable of moving at different speeds depending on which muscles are involved. Larger muscles in the arm generally produce slower, broader movements compared to the smaller muscles in the fingers, which allow for faster, finer motions. The following section uses frequency analysis, including *power spectral density* (PSD) and *spectrograms*, to characterize different types of hand motions that generate distinct frequency components.

3.2.1 Power Spectral Density

Power spectral density (PSD) quantifies the power present in various frequency components of a time-series signal. It is particularly useful in identifying dominant frequencies within a signal and understanding how energy is distributed across different frequency bands. PSD is calculated using the fast Fourier transform (FFT), which decomposes a signal into its constituent frequencies, allowing for the analysis of the signal's frequency content [15]. The PSD is estimated using the FFT, based on the formula:

$$S(f) = \lim_{T \to \infty} \frac{1}{T} \left| \mathcal{F}\{x(t)\}(f) \right|^2, \tag{3.1}$$

where $S(f)$ is the Power Spectral Density, $\mathcal{F}\{x(t)\}$ represents the FFT of the signal $x(t)$, T is the duration of the signal, and f is the frequency.

3.2.2 Spectrogram

A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. Unlike PSD, which provides a static view of the frequency content, spectrograms offer a time-varying perspective, showing how the spectral density of the signal changes over time [15]. This is particularly useful for analyzing signals with non-stationary or evolving frequency components, such as hand motions. The spectrogram is calculated using the FFT of successive time-windowed segments of the original signal:

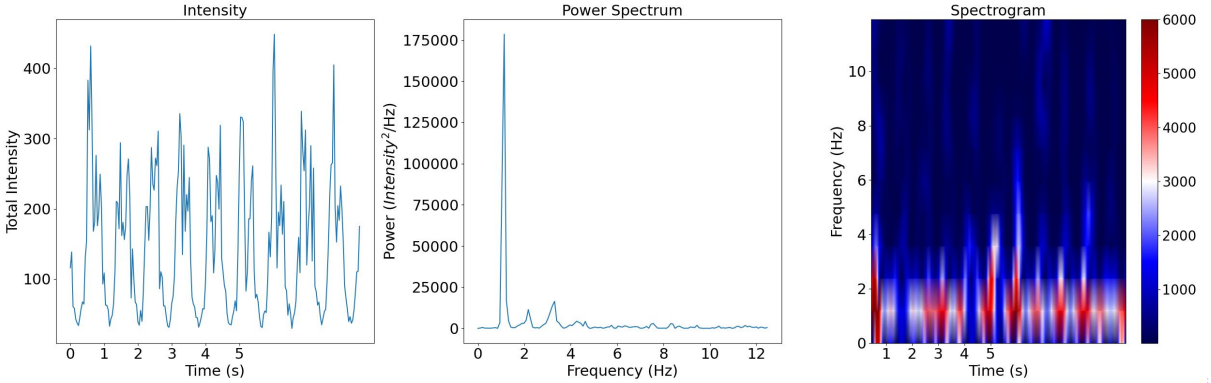
$$S(t, f) = |\mathcal{F}\{x(t)w(t)\}(f)|^2, \tag{3.2}$$

where $S(t, f)$ represents the spectrogram, $\mathcal{F}\{x(t)w(t)\}(f)$ is the FFT of the signal $x(t)$ multiplied by a window function $w(t)$, centered around time t, and f is the frequency.

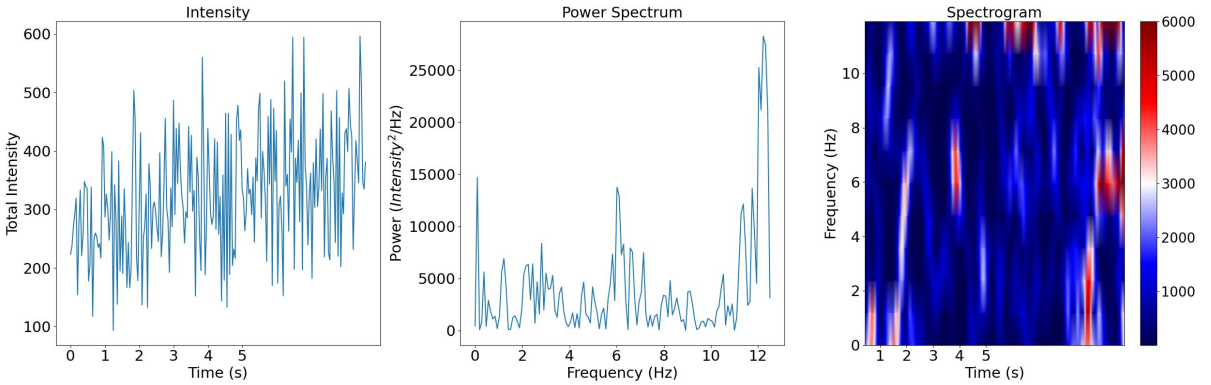
3.2.3 Spectral Profiles of Low and High-Frequency Hand Motions

Not all hand motions exhibit the same frequency characteristics, as they vary widely depending on the activity and intent. The following analysis categorizes hand motions into two types based on their frequency components.

1. **Low-Frequency Hand Motions:** These hand motions are slower movements that occur either intentionally or unintentionally during day-to-day activities. These motions are typically generated by larger muscles, often involving the whole arm, and result in broader, movements. For example, gestures made while speaking, hand motions while cooking, such as stirring a pot, or arms swinging back and forth while walking. These motions could be rhythmic or non-rhythmic and they result in high power at low frequencies which generally occur in the 1-4Hz range. Figure 3.1a illustrates the frequency analysis (signal intensity (left), PSD (middle) and spectrogram (right)) of a 10-second recording as a hand swings back and forth in



(a) Spectral profile of an arm swinging. The signal PSD, and spectrogram reveal a dominant low-frequency peak between 1-2Hz, reflecting the rhythmic swinging motion of the arm.



(b) Spectral profile of rapid finger movements. The PSD and spectrogram show high-frequency peaks between 6-12Hz, indicating rapid, rhythmic oscillations of the fingers.

Figure 3.1: Signal intensity (left), PSD (middle), and spectrogram (right) for *low-frequency* and *high-frequency* hand motions detected by Soli over a 10-second period.

front of the Soli. In the PSD, a large and sharp peak is clearly visible around 1-2Hz. This peak represents the primary frequency of the back and forth swinging motion of the hand. The high power of this peak also suggests a rhythmic and repetitive motion. Its location at 1-2 Hz implies that the hand was swinging back and forth approximately every 0.5-1 second. After this dominant peak, there's a notable drop in power, emphasizing the significant contrast between the primary motion and any other minor activities or noises. In the spectrogram, the bright bands that are persistent at around 1-2 Hz throughout the 10-second span further demonstrate the continuous presence of the primary swinging motion.

2. **High-Frequency Hand Motions:** These hand motions involve rapid, high-speed movements that are rhythmic in nature. These actions demand conscious and precise

effort, which means they rarely happen by chance or without intent. The rhythmic repetition of these motions generates strong power in the 4-12 Hz frequency range. One example of such a motion is repeatedly tapping the tip of the thumb with the index finger in a quick, oscillating pattern. These types of gestures involve fine motor control and small displacements, yet they generate distinct, high-frequency signatures due to the speed and regularity of the motion. Figure 3.1b presents the frequency analysis of a 10-second recording taken while rapidly moving fingers in front of the Soli. Compared to the PSD in Figure 3.1a, there are many more pronounced peaks at higher frequencies, due to the fast, rhythmic motions of individual fingers. The big peak between 10-12Hz indicates a dominant frequency in the finger motions, which suggests that one or multiple fingers had a specific oscillatory pattern that repeated roughly every 0.08 to 0.1 seconds. In the spectogram, the bright bands of high-frequency activity in the 6-12 Hz range reaffirm the high-frequencies generated by the rapid finger motions throughout the recorded period.

This analysis indicates that while day-to-day hand motions encompass a variety of movements, they typically do not produce significant high-frequency components in the signal. Routine activities such as cooking, walking, or gesturing while talking predominantly generate low-frequency signals with peaks in the 1-4 Hz range. In contrast, hand motions that produce strong power in the higher frequency bands (4-12 Hz) require intentional, high-speed, and rhythmic movements. These high-frequency motions are not performed accidentally during normal daily tasks but are associated with deliberate actions involving rapid and repetitive movements. While this section presents only two representative examples—one low-frequency and one high-frequency—it serves to build intuition about the relationship between motion characteristics and their frequency profiles. A broader range of hand and finger motions will be examined later in the chapter.

3.3 High-Frequency Subtle Activation Gestures

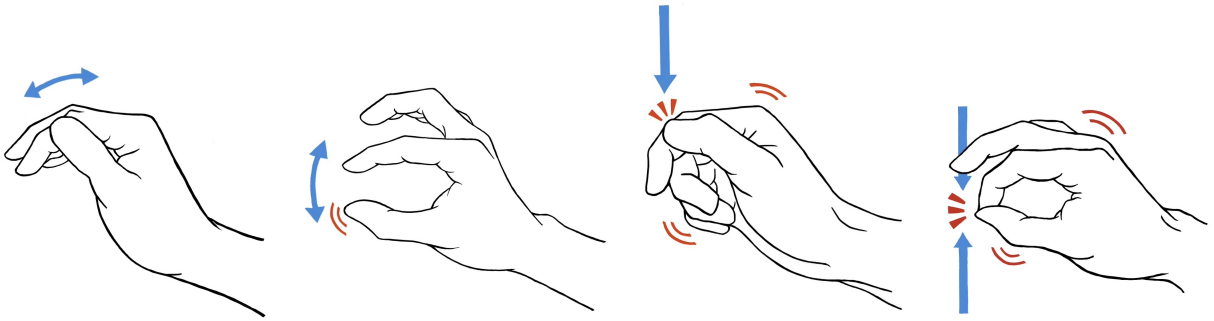
As previously mentioned, activation gestures need to be distinct from everyday hand motions. Hand motions that produce strong power in high-frequency ranges (4-12Hz) are well-suited for this purpose because their rhythmic and high-speed nature makes them less likely to occur by accident. *Rhythmic gestures* are intentional, repeated movements where the motion pattern remains consistent over time [38, 22, 23]. The deliberate and rhythmic nature of high-frequency gestures ensures that they are distinct from slower and often non-repetitive motions common in everyday activities.

The following sections will describe four candidate rhythmic activation gestures: *Finger Taps*, *Finger Rubs*, *Thumb Presses*, and *Pinch Presses*. The selection of these four gestures was guided by both technical and design-driven considerations. All gestures were chosen for their potential ability to generate high-frequency components within the 4–12 Hz range. Beyond this, the gestures align with qualities associated with subtle gestures—specifically, those emphasizing spatial compactness, low physical exertion, and social unobtrusiveness. As discussed in Section 2.6.2, subtle gestures often overlap with the notion of micro-gestures: small-scale, low-amplitude hand or finger motions [10, 92]. By keeping the movements localized to the fingers and minimizing arm displacement, the selected gestures preserve these characteristics while still producing radar-detectable high-frequency signals.

Preliminary recordings of each gesture were conducted by the researcher using a Soli radar to examine whether these gestures indeed produce expected high-frequency components. The gestures have been grouped into two categories based on the nature of the motions involved: *Rapid Finger Gestures* and *Tremor-Inducing Pressure Gestures*.

3.3.1 Rapid Finger Gestures

These gestures are characterized by rapid finger movements, primarily involving the thumb and index finger. The two *Rapid Finger Gestures* are:



(a) Finger Rubs: The thumb and index finger are rubbed together quickly in a back-and-forth motion.

(b) Finger Taps: The fingertip of the index finger rapidly taps against the thumb in a repeated motion.

(c) Thumb Press: The thumb repeatedly applies firm pressure against the radial side of the index finger.

(d) Pinch Press: The thumb and index finger repeatedly pinch together, applying firm pressure.

Figure 3.2: Candidate subtle activation gestures.

1. **Finger Rubs**: This gesture (shown in figure 3.2a) involves the back and forth rubbing of the thumb and index finger. Gently rubbing these two fingers together quickly and repeatedly can generate high-frequency signals, as demonstrated in the spectral analysis in Figure 3.3a. The analysis shows rapid fluctuations in intensity and a pronounced peak in the PSD around the 10-12Hz range. The spectrogram further illustrates strong activity at these frequencies, highlighting the generation of strong high-frequency components.
2. **Finger Taps**: This gesture (shown in figure 3.2b) involves quick, repetitive motions where the fingertip of the index finger swiftly taps the thumb. Rapidly tapping the index finger against the thumb in this manner can produce high-frequency components, as demonstrated in the spectral analysis in Figure 3.3b. This gesture exhibits some low-frequency peaks in the PSD within the 0-4Hz range. However, more prominent peaks are visible in the 8-12Hz range. Additionally, the spectrogram also highlights bright bands at these frequencies, demonstrating the high-frequency energy generated by this gesture.

Variations of both these gestures have been featured in research from Google and Infineon Technologies [83, 30]. Additionally, similar gestures have been used in gesture recognition research using other types of sensing technologies, such as accelerometers and gyroscopes

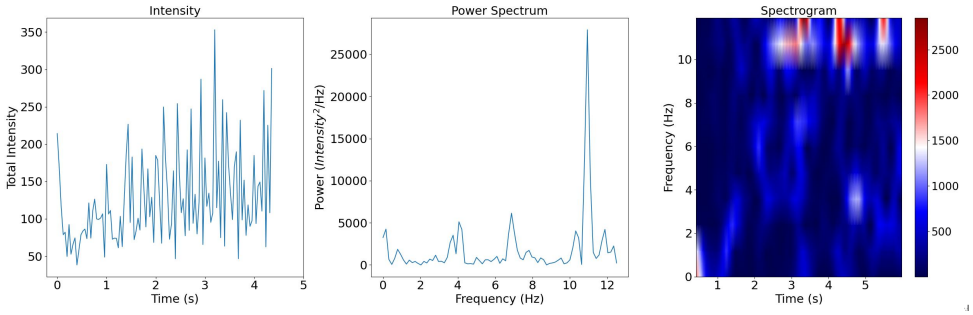
in smartwatches [88], capacitive sensors in data gloves [53], and even infrared sensors [25]. In this work, *Finger Rubs* and *Finger Taps* have been adapted to specifically induce high-frequency components through rhythmic repetition. Generating strong spectral energy within the 4–12 Hz range would require performing these gestures at a repetition rate of approximately 3–6 times per second.

3.3.2 Tremor-Inducing Pressure Gestures

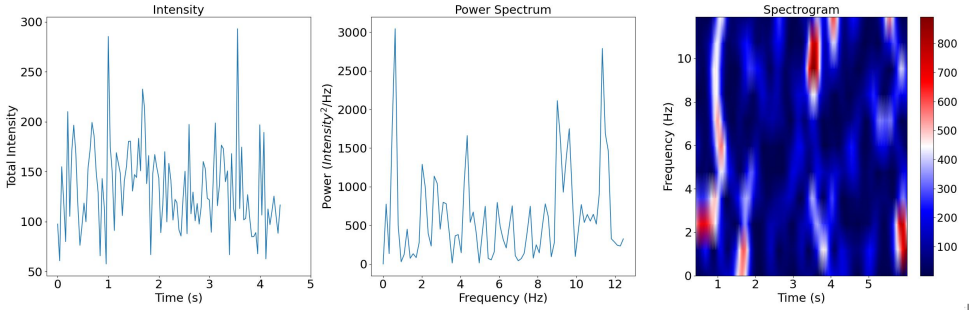
These gestures generate *isometric tremors*. Isometric tremors are induced by the consistent, deliberate application of pressure, leveraging the natural response of muscle contractions [20]. The two *Tremor-Inducing Pressure Gestures* are:

1. **Thumb Presses**: This gesture (shown in figure 3.2c) involves firmly and repeatedly pressing the radial side of the index finger using the thumb. This action creates a concentrated pressure on the joint which engages muscles in the hand and forearm resulting in a localized isometric tremor that leads to the generation of high-frequency components as can be seen in the spectral analysis in Figure 3.3c. The intensity plot shows frequent spikes corresponding to the repeated application of pressure. In the PSD, there are peaks between 8-12 Hz, indicative of the high-frequency components generated by the tremors. The spectrogram further validates this, displaying bright bands at these frequencies.
2. **Pinch Presses**: This gesture (shown in figure 3.2d) involves firmly and repeatedly pinching the thumb and index finger together, thereby inducing an isometric tremor that leads to the generation of high-frequency components, as can be seen in the spectral analysis in Figure 3.3d. The intensity plot exhibits frequent spikes corresponding to the application of pressure, and the PSD shows peaks around 6-8Hz, highlighting the high-frequency components generated by the tremors. The spectrogram also displays bright bands at these frequencies.

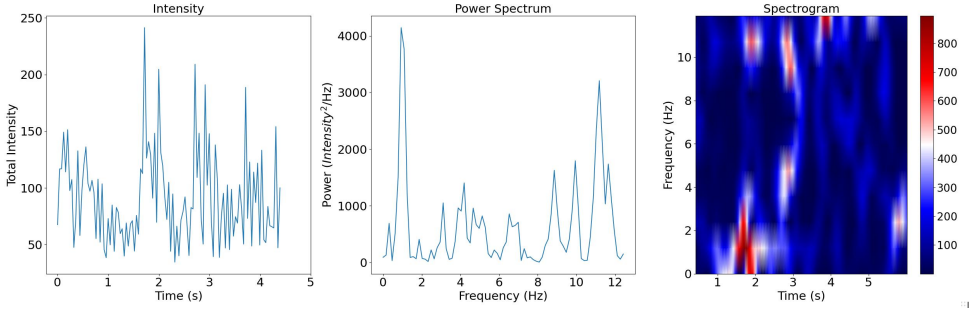
These two gestures were selected due to their ability to generate subtle isometric tremors



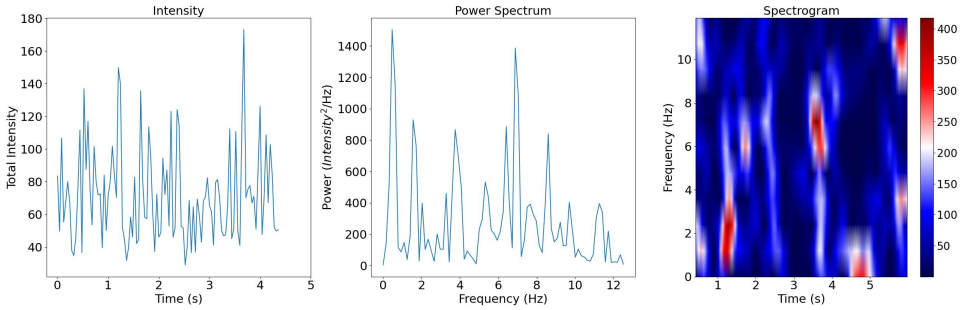
(a) Spectral profile of the *Finger Rubs* gesture, showing high-frequency components primarily in the 10-12 Hz range generated by rapid thumb and index finger rubbing.



(b) Spectral profile of the *Finger Taps* gesture, highlighting high-frequency signals in the 8-12 Hz range produced by quick, repetitive tapping between the thumb and index finger.



(c) Spectral profile of the *Thumb Presses* gesture, displaying high-frequency tremors in the 8-12 Hz range caused by repeated pressure on the radial side of the index finger.



(d) Spectral profile of the *Pinch Presses* gesture, showing high-frequency tremors in the 6-8 Hz range generated by repetitive pinching between the thumb and index finger

Figure 3.3: Spectral profiles of activation gestures using Soli over a 5-second period. Each plot shows the intensity (left), power spectral density (PSD) (middle), and spectrogram (right), illustrating the high-frequency components induced by each gesture.

through deliberate pressure application. Unlike *Rapid Finger Gestures* that involve visible finger movement, these gestures require the hand to remain in a fixed pose while repeatedly applying pressure. The absence of noticeable finger motions makes them highly subtle, but also more challenging to detect. Similar to the Rapid Finger Gestures, these pressure-based gestures would also need to involve applying pressure approximately 3–6 times per second to generate strong spectral energy within the 4–12 Hz range.

3.4 Activation Gesture Recognition Pipeline

Since the previous section demonstrated that the proposed candidate activation gestures are capable of producing high-frequency components, the next step involves developing a method to recognize these gestures. The following sections will outline the design of a pipeline that processes sensor data to recognize subtle activation gestures.

3.4.1 Pipeline Components

Figure 3.4 gives an overview of the proposed *activation gesture recognition pipeline*. This pipeline leverages the power distribution within frequency bands to differentiate intentional activation gestures from incidental hand motions. The various stages of the pipeline are detailed below, and the pseudocode for implementation is outlined in Algorithm 1.

1. **Total Intensity Calculation:** The pipeline begins by collecting raw sensor data, which is processed to derive the intensity of the hand motions. The focus is on deriving the overall signal intensity as detected by the sensor. For instance, in the case of the Soli, calculating the total intensity means taking the absolute sum of the range-Doppler map (RDM). This intensity serves as the basic input for the subsequent stages of the pipeline. The intensity data is updated in a sliding window buffer that retains the most recent 1-second period of motion data. This window length was

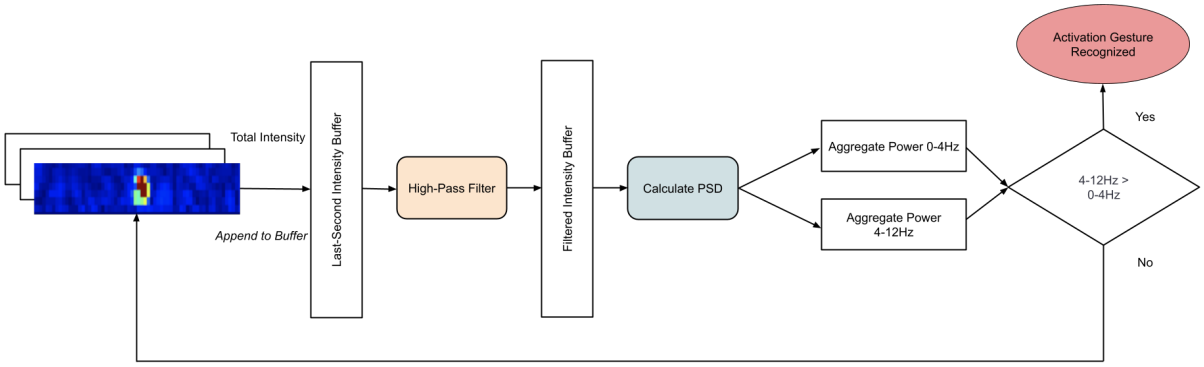


Figure 3.4: Overview of the proposed *activation gesture recognition pipeline*. First, total intensity is calculated and stored in a 1-second sliding window buffer. A high-pass filter then suppresses low-frequency components, after which the PSD is calculated. The PSD is aggregated into two frequency bands: 0-4Hz for low-frequency motions and 4-12Hz for high-frequency gestures. The system recognizes an activation gesture if the power in the 4-12Hz band exceeds that in the 0-4Hz band.

empirically chosen to balance frequency resolution and system responsiveness: a shorter window would reduce detection latency but make it harder to distinguish frequency bands reliably, while a longer window would improve frequency resolution at the cost of increased lag. A 1-second window provides sufficient resolution.

2. **Signal Filtering:** Filtering is used in gesture recognition systems for isolating desired gesture signals from noise. Sometimes sensor data can be dominated by low-frequency components, which can obscure the subtle, high-frequency gestures. However, these low-frequency components are not merely noise—they reflect broad, casual hand motions that the system must be able to detect and distinguish from intentional input. To achieve this, the pipeline applies a high-pass Butterworth filter to the buffered intensities. Rather than fully suppressing low-frequency content, the filter attenuates its power just enough to reduce its dominance, thereby allowing the high-frequency components to emerge more clearly when performing a subtle activation gesture. For instance, in the case of the *Finger Taps* gesture (Figure 3.3b), low-frequency peaks in the 0-4 Hz range are present. Similarly, in *Thumb Presses* (Figure 3.3c) and *Pinch Presses* (Figure 3.3d), low-frequency peaks can also be observed. The high-pass filter

reduces the influence of these low-frequency components just enough to prevent them from overshadowing high-frequency gesture signals, while still retaining sufficient low-frequency information to reflect broader, casual hand motions.

3. **Calculate PSD:** In this step, the PSD of the filtered intensities is computed. This gives the distribution of energy across different frequencies in real-time, thus providing the spectral characteristics of the ongoing gesture.

4. **Aggregate Power in Frequency Bands:** The PSD is then aggregated into two separate frequency bands: 0-4Hz and 4-12Hz. The 0-4Hz band captures the total power of low-frequency components. As discussed in Section 3.2.3, day-to-day hand motions generate significant power in this band as they tend to be low-speed and typically non-rhythmic. The 4-12Hz band captures the total power of the high-frequency components, which exhibit higher power when the deliberate, rhythmic activation gestures are performed, as explored in Section 3.3.

5. **Gesture Recognition:** The final stage of the pipeline uses the aggregated powers to determine whether the motion detected is an activation gesture. If the power in the 4-12Hz band is greater than in the 0-4Hz band, the recognition condition is met, indicating that the hand gesture being performed has strong high-frequency components. This suggests that the gesture is fast and rhythmic, characteristic of the candidate activation gestures. On the other hand, if the power in the 0-4Hz band is dominant, it indicates the presence of stronger low-frequency components, associated with broader, slower motions.

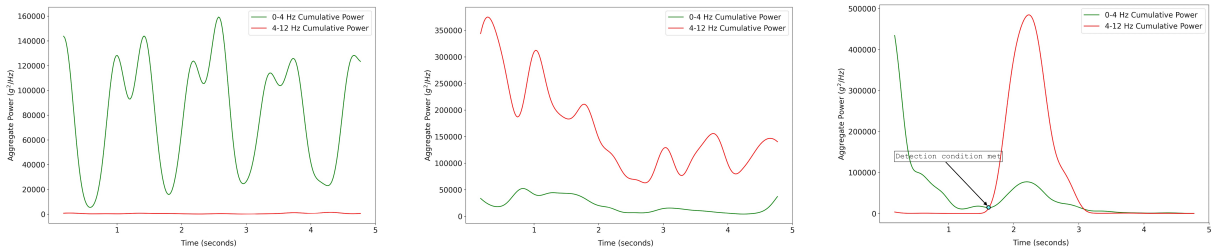
Data: Sensor data stream, sliding window size (history_length for 1 second), high-pass filter parameters, frequency bands (0-4 Hz, 4-12 Hz)
Result: Detected activation gestures
begin
  Initialize *total_intensities* deque with max length *history_length*
  Initialize $filtered\_intensities$ deque with max length *history_length*
  while *running* **do**
    Retrieve raw data from sensor stream
    Process raw data to calculate *signal_intensity* representing motion intensity
    Append *signal_intensity* to *total_intensities*
    if *total_intensities has sufficient data for 1 second* **then**
      $filtered\_data \longleftarrow$ Apply high-pass filter to *total_intensities*
      $(low\_frequency\_power, high\_frequency\_power) \longleftarrow$ Calculate PSD and aggregate power in frequency bands
      if $high\_frequency\_power > low\_frequency\_power$ **then**
        Activation gesture recognized
        Clear *total_intensities* and $filtered\_intensities$
      end
    end
  end
end

Algorithm 1: Pseudocode for the proposed *activation gesture recognition pipeline*

3.4.2 Visual Analysis

Figure 3.5 shows three graphs, each tracking the cumulative power in the low (0-4Hz) and high (4-12Hz) frequency bands during a 5-second window of different hand motions as recorded by the Soli. The plots in 3.5a shows the cumulative low and high frequency powers during a typical non-gesture event: an arm swinging. It exhibits dominant power in the lower frequency band and shows no significant spikes in the high-frequency band, reflecting the absence of fast, rhythmic motions. The plots in 3.5b show the cumulative low and high frequency powers when continuously performing the *Finger Rubs* activation gesture. Unlike the arm swing, the cumulative high-frequency power is consistently higher than the cumulative low-frequency power due to the rapid, rhythmic finger motions, which generate strong high-frequency components in the signal. This shows the pipeline's ability to effectively track the power in high-frequency components during the activation gesture while adequately attenuating the low-frequency components. Figure 3.5c illustrates the critical moment when an activation gesture (*Finger Rubs*) is recognized. Here, the

cumulative high-frequency power exceeds the cumulative low-frequency power, triggering the detection condition.



(a) Cumulative low and high-frequency powers during a non-gesture arm swing, with dominant low-frequency power due to slow, broad motions.

(b) Cumulative low and high-frequency powers during the *Finger Rubs* gesture, showing consistently higher high-frequency power due to fast, rhythmic motions.

(c) Cumulative high-frequency power exceeding cumulative low-frequency, marking the activation gesture trigger condition.

Figure 3.5: Graphs tracking cumulative power in low (0-4Hz, green) and high (4-12Hz, red) frequency bands during a 5-second period of different hand motions as captured by the Soli: (a) arm swinging back and forth, (b) continuous *Finger Rubs* gesture, and (c) activation gesture trigger point.

3.5 Evaluation

Since the input to the pipeline is signal intensity, the system is sensor-agnostic and can be adapted to any sensor capable of detecting hand motion. To demonstrate the pipeline's adaptability and assess its performance across different sensing modalities, a user study was conducted using three types of sensing technologies: a Google Soli radar, an Intel D435 camera augmented with MediaPipe hand tracking [97], and a wrist-mounted SHAKE [1], which is a Bluetooth enabled sensor that includes an accelerometer. The study focuses on testing the following hypotheses:

H1: *The pipeline effectively ignores casual hand movements.*

This hypothesis directly addresses the Midas Touch problem and evaluates a core functionality of the pipeline: its ability to filter out casual, everyday hand movements.

[1] `https://code.google.com/archive/p/shake-drivers/`

H2: *The pipeline recognizes Rapid Finger Gestures (Finger Rubs, Finger Taps) faster than Tremor-Inducing Pressure Gestures (Thumb Press, Pinch Press).*

This hypothesis aims to evaluate whether *Tremor-Inducing Pressure Gestures*, being subtler due to their reliance on muscle tremors, take longer to recognize compared to the more pronounced and intentional movements of *Rapid Finger Gestures*.

H3: *The SHAKE recognizes activation gestures faster than the other sensors.*

This hypothesis investigates the impact of sensor characteristics on recognition speed, with the expectation that the SHAKE's high sampling rate (see Table 3.1) and wrist-mounted configuration will enable faster recognition of the activation gestures compared to the other sensors.

H4: *Users prefer Rapid Finger Gestures over Tremor-Inducing Pressure Gestures.*

This hypothesis investigates whether users will prefer *Rapid Finger Gestures* over *Tremor-Inducing Pressure Gestures*, considering factors such as overall comfort during repeated use and recognition performance.

3.5.1 System Description

Sensor Sampling Rates: As mentioned before, the study uses three different sensors, each with unique sampling rates. According to the Nyquist-Shannon sampling theorem, to get the complete gesture signal, the sampling frequency must be at least twice the highest frequency present in the gesture signal [67]. The specific sampling rates and the corresponding maximum detectable frequencies for each sensor are detailed in Table 3.1. The sampling requirements are met by all sensors, as their sampling rates are sufficiently high to detect frequencies in casual movements and the candidate subtle activation gestures.

Sensor	Sampling Rate (Hz)	Max Detectable Frequency (Hz)
Google Soli Radar	25	12.5
Intel D435 Camera	30	15
SHAKE Accelerometer	60	30

Table 3.1: Sampling specifications of sensors.

Processing Sensor Input Data : The signal intensity used as input for the pipeline is processed differently for each sensor. The following steps outline how the raw data from each sensor is transformed into a single intensity value at each timestep.

- **Google Soli:** At each timestep, Soli provides three RDM outputs corresponding to its three receiver antennas. As discussed in Section 2.4.5 each RDM embeds the range and velocity of the gesture as captured by the corresponding antenna. To compute the intensity value at each timestep, the absolute sum of all RDMs is calculated.
- **Intel D435 Camera with MediaPipe Tracking:** MediaPipe Hands provides tracking of 21 different 3D landmarks on the hand [97]. In this study, only the x and y coordinates are used, as these are sufficient for tracking the horizontal and vertical gesture motions. At each timestep, the speed of each landmark is calculated and combined to create a composite intensity value that represents the overall speed of the hand.
- **SHAKE Accelerometer:** At each timestep, the absolute values of acceleration along the x, y, and z axes are summed to form the intensity value. This represents the magnitude of acceleration.

Feedback System: To provide participants with real-time guidance on their gestures, a feedback system employing visual and auditory mechanisms was developed. Figure 3.6 gives a graphical representation of this system. In the center is a vibrating circle that dynamically responds to the participant's gestures. The circle's size and vibration are controlled by adjusting its radius, which reflects both the vibration amplitude and frequency. These properties are determined by the cumulative power in the high-frequency band (4-12Hz) and the ratio of the cumulative power in the high-frequency band to the cumulative power in the low-frequency band (0-4Hz) in the input signal. Consequently,

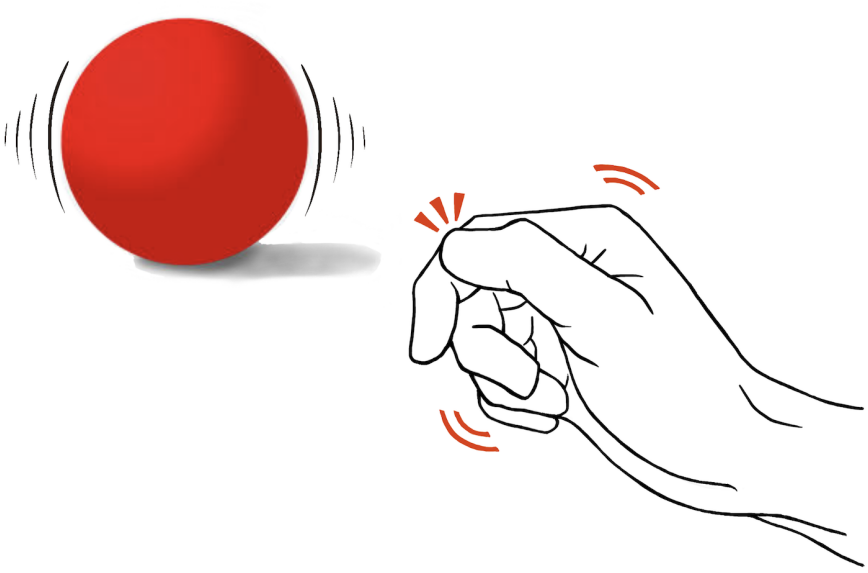
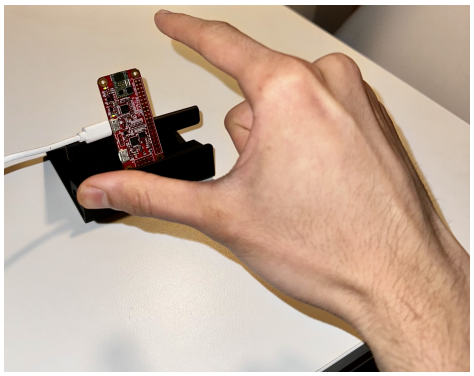


Figure 3.6: Feedback system used to test the activation gesture pipeline. The system features a vibrating circle that responds to the participant's gestures, adjusting its size and vibration based on the gesture's speed and intensity. Faster gestures increase vibration frequency, while strong signal intensities result in more visible vibrations. The circle turns green and emits a beep when a gesture is successfully recognized.

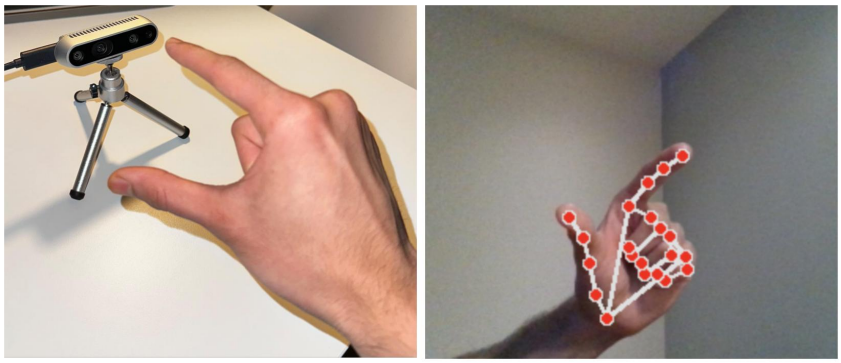
for gestures like *Finger Taps* or *Finger Rubs*, performing the gesture with faster rhythmic motions results in the circle vibrating at a higher frequency. Also, if the intensity of the detected signal is higher, the amplitude of the circle's vibration will increase, making it pulsate more visibly. Similarly, for gestures like *Thumb Presses* or *Pinch Presses*, applying pressure more frequently increases the vibration frequency. Additionally, applying greater pressure generates more pronounced tremors, causing the circle to vibrate more visibly. Once the activation condition is met—when the power in the high-frequency band exceeds that of the low-frequency band—the system emits a beep, and the circle changes color to green, signaling that the gesture has been recognized. Users then need to wait until the circle turns red again before attempting another gesture.

3.5.2 Procedure

Participants were seated comfortably in front of a laptop. Two sensors, the Google Soli radar and the Intel D435 webcam (augmented with MediaPipe hand tracking), were directly



(a) Google Soli radar setup with the sensor positioned 20cm from the table edge to ensure gestures are performed within the sensing range.



(b) Intel D435 camera setup (left) with real-time MediaPipe hand tracking visualized in an OpenCV window (right).

(c) SHAKE sensor pack (left) worn on the wrist (right).

Figure 3.7: Experimental setups for the activation gesture user study.

connected to the laptop. For the SHAKE accelerometer, a USB dongle was attached to the laptop, which connected to the SHAKE sensor pack wirelessly via Bluetooth. Sensor-specific setup and instructions were as follows:

- **Google Soli Radar:** Participants needed to perform the gestures within Soli's sensing range of 20cm. To ensure this, the Soli was positioned 20cm from the edge of the table, and participants were instructed to perform the gestures within the edge of the table area, keeping their hands within the sensor's detection zone (shown in Figure 3.7a).
- **Intel D435 Webcam with MediaPipe:** An OpenCV window displayed the hand with MediaPipe's hand tracking, and participants were instructed to keep their hands within the visible frame of the webcam while performing gestures (shown in Figure 3.7b).
- **SHAKE Accelerometer:** The SHAKE was secured to the participant's wrist with a wrist band (shown in Figure 3.7c). This placement was chosen to simulate the use of smartwatches, which commonly incorporate accelerometers. Additionally, the muscles around the wrist are engaged while performing the candidate activation gestures. Participants were instructed to keep their arm in a naturally bent position while performing the gestures.

The order in which the sensors were chosen was randomized to account for order effects, helping to prevent bias in the results due to the sequence in which the sensors were tested. Sensors were tested individually rather than simultaneously, as doing so would have required participants to keep their hand within both the Soli's limited 20cm sensing range and the MediaPipe OpenCV tracking window—an impractical constraint during gesture performance. Additionally, testing sensors individually ensured that each sensor had an optimal view of the participant's hand, not only during gesture performance but also when performing casual hand movements. Once a sensor was selected, one of the four candidate gestures was randomly chosen, and participants were required to perform each gesture 20 times per sensor. Real-time feedback was provided through the vibration system

described previously. The order of gestures was semi-randomized, alternating between a *Rapid Finger Gesture* and a *Tremor-Inducing Pressure Gesture*. This alternation was done to prevent the participant's hand from tiring due to repeatedly performing the same type of gesture. To assess the ability of the pipeline to filter out casual movements, participants were asked to perform a series of tasks for 4 minutes with each sensor. This session was placed after performing two activation gestures with that sensor, providing a break to prevent hand fatigue. The tasks included: browsing a phone in front of the sensor (or while wearing the SHAKE), performing broad hand motions, writing on a piece of paper, and typing on a keyboard. Finally, at the end of the study, participants were asked to rank the gestures in order of preference and share the reasons for their choices.

3.5.3 Metrics

The following metrics were used to assess the system:

- **False Activations:** This refers to the number of times the activation condition was met while participants were performing the set of casual hand movements. Any trigger that occurred during these movements was recorded as a false activation.
- **Time to Activation:** This is the time taken for a gesture to be recognized after the participant begins performing it. The timer was initiated as soon as the red circle turned green—signaling that the system was ready for a new input—and stopped when the activation condition was met.
- **Gesture Rankings:** This is the participants' subjective ranking of the four activation gestures.

3.5.4 Participants

Eight participants were recruited for the study, consisting of six males and two females, with ages ranging from 24 to 36 years (average age: 27). All participants performed the experiment using their right hand. Ethics approval for this study was provided by the

university's ethics committee.

3.5.5 Results

The results are described below with respect to each specific hypothesis.

H1: *The pipeline effectively ignores casual hand movements.*

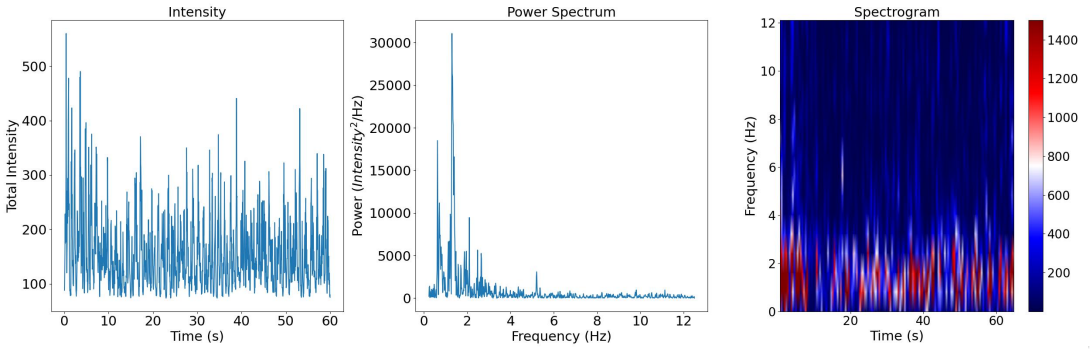
Table 3.2 provides a summary of false activations recorded when participants performed different activities with the three sensors. The pipeline had zero false activations for broad hand gestures across all sensors, indicating that the system is effective at filtering out broad, low-speed motions. However, for activities like typing, writing, and phone usage, there were false activations across all sensors.

Sensor	Broad Gestures	Typing	Writing	Phone Usage	Total
Soli	0	12	3	6	21
Intel D435	0	16	23	35	74
SHAKE	0	32	29	21	82

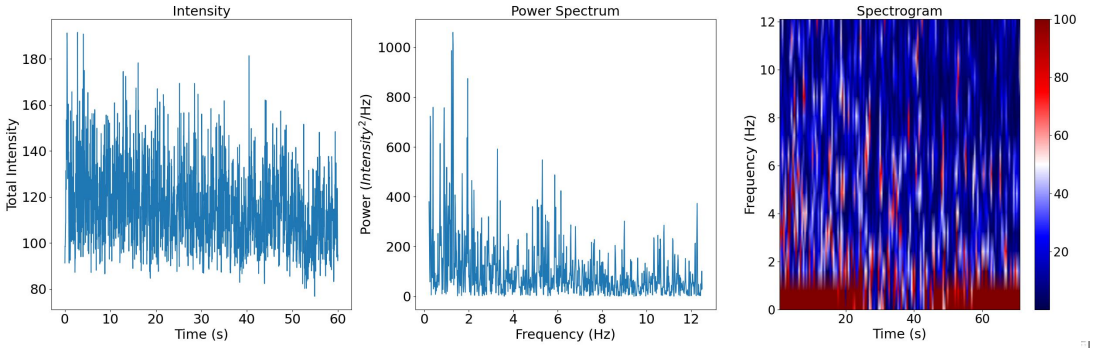
Table 3.2: Summary of false activations by sensor and activity

Spectral analysis was performed to understand these results. For each sensor and activity, signal intensities from all participants were aligned and averaged at each time point to form a composite signal. The spectral profiles of these composite signals were then produced to examine the frequency components associated with each activity. Figure 3.8 shows the spectral profiles for activities recorded with the Soli. Spectral profiles for the Intel D435 and SHAKE sensors are included in Appendix A.1.

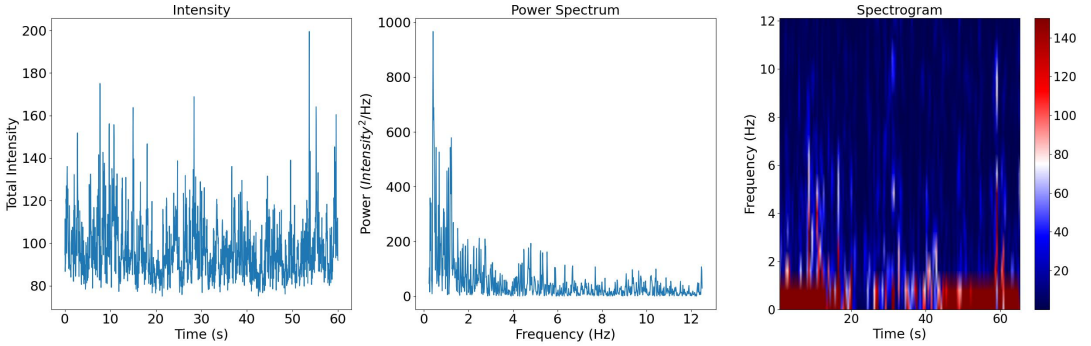
In Figure 3.8a, the power spectrum for broad hand motions reveals strong peaks in the low-frequency range (0-4Hz), with weak power in the higher frequencies. This pattern is also seen in the spectrogram, where the bright bands of activity are concentrated below 4Hz. The same trend was observed with the Intel D435 and SHAKE sensors, where broad hand motions primarily exhibited strong power below 4Hz. The lack of strong



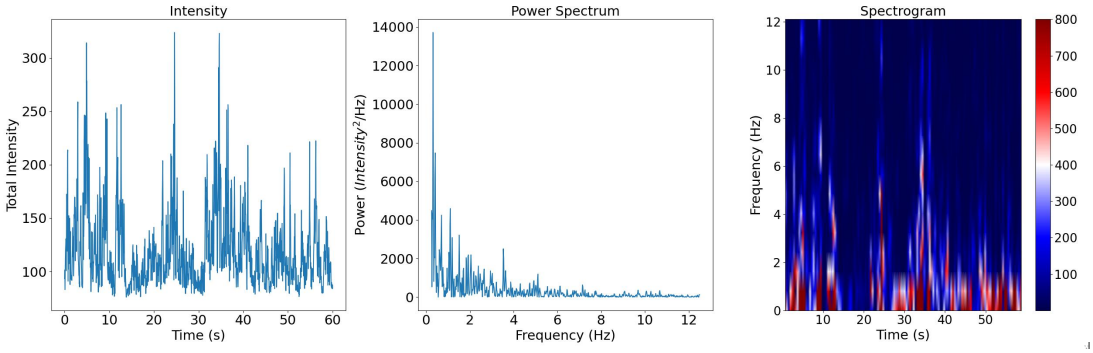
(a) Spectral profile of broad hand gestures, showing strong low-frequency components primarily in the 0-2Hz range, with minimal high-frequency activity.



(b) Spectral profile of typing, showing high-frequency components in the 4-12Hz range generated by rapid finger movements.



(c) Spectral profile of writing, showing dominant low-frequency activity below 4Hz with occasional high-frequency bursts, likely arising from fast writing speeds and adjustments.



(d) Spectral profile of phone usage, primarily showing strong low-frequency activity below 4 Hz, with sporadic high-frequency peaks likely arising from quick taps or swipes on the screen.

Figure 3.8: Spectral profiles of various activities recorded with the Soli, generated from composite signals created by aligning and averaging signal intensities across all participants.

high-frequency components in broad hand motions allowed the system to effectively ignore these movements, resulting in zero false activations across all sensors for this activity.

The power spectrum for typing (Figure 3.8b) shows high-frequency components, with strong peaks between 4-8Hz. This is also visible in the spectrogram, where bright bands appear intermittently at higher frequencies. These high-frequency components are produced from the fast finger movements involved in typing, which triggered the activation condition resulting in false activations. The power spectrums for both writing (Figure 3.8c) and phone use (Figure 3.8d) predominantly shows strong power below 4Hz, similar to broad hand motions. Most of the activity is concentrated in this range, as seen by the peaks in frequencies under 4Hz and the bright, continuous bands in the lower part of the spectrograms. However, there are occasional, sporadic bursts of high-frequency activity. In the case of writing, these high-frequency components likely arise from fast writing speeds and adjustments. Similarly, in phone usage, high-frequency bursts likely occurred during rapid taps or swipes on the screen.

Multi-Trigger Validation: To address the issue of false activations during typing, writing, and phone usage, a *post-hoc* multi-trigger validation mechanism was implemented. This approach requires multiple activation signals within a brief time window to confirm an intentional activation. The idea is that casual movements are less likely to meet a repeated threshold, so this method could help distinguish intentional high-frequency activation gestures from other high-frequency hand motions that arise from activities like typing or writing.

A 2-second time window was selected, and validation criteria were set at 2 and 3 triggers within this window. The modified algorithm of the *activation gesture recognition pipeline* with the multi-trigger condition is included in the appendix in Section A.2. Figure 3.9 shows the impact of applying these multi-trigger criteria. The false activations decreased significantly with the 2-trigger condition and under the 3-trigger criterion, false activations were completely eliminated for all activities except typing with the SHAKE sensor, which

still had 3 false activations.

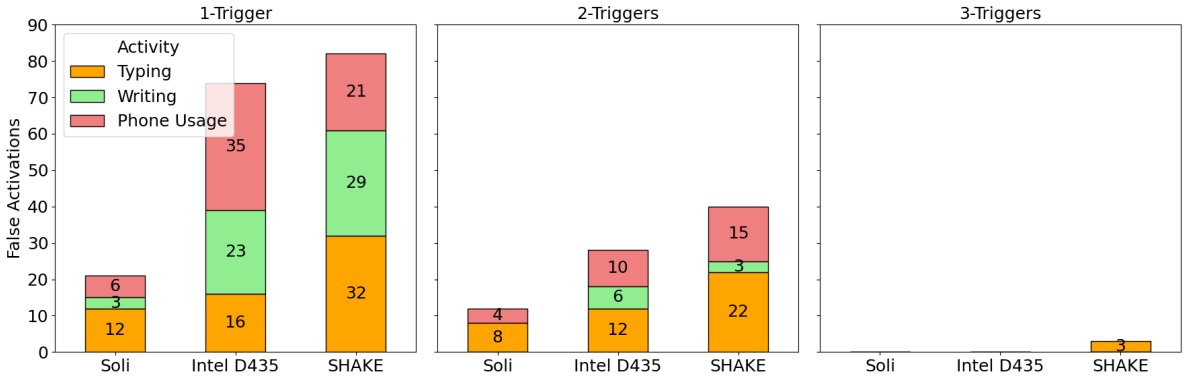


Figure 3.9: Effect of multi-trigger validation on false activations across different sensors and activities. The figure compares the number of false activations under 1-trigger, 2-trigger, and 3-trigger criteria within a 2-second window. The 2-trigger criterion significantly reduces false activations, while the 3-trigger criterion nearly eliminates them across all sensors and activities, except for a small number of activations during typing with the SHAKE sensor.

H2: *The pipeline recognizes Rapid Finger Gestures (Finger Rubs, Finger Taps) faster than Tremor-Inducing Pressure Gestures (Thumb Press, Pinch Press).*

To first test whether there is a significant difference in time to activation between *Rapid Finger Gestures* and *Tremor-Inducing Pressure Gestures*, an analysis involving normality testing and non-parametric comparisons was conducted. The Shapiro-Wilk test was employed to assess the normality of the time to activation data for both types of gestures across all three sensors. The results indicated that none of the distributions conformed to normality ($p \leq 0.01$).

To first test whether there is a significant difference in time to activation between *Rapid Finger Gestures* and *Tremor-Inducing Pressure Gestures*, an analysis involving normality testing and non-parametric comparisons was conducted. The Shapiro-Wilk test was employed to assess the normality of the time to activation data for both gesture types across all three sensors. The results indicated that none of the distributions conformed to normality: Soli (*Rapid Finger Gestures:* $W = 0.81$, $p < 0.001$; *Tremor-Inducing Pressure Gestures:* $W = 0.77$, $p < 0.001$), Intel D435 ($W = 0.82$, $p < 0.001$; $W = 0.85$, $p < 0.001$),

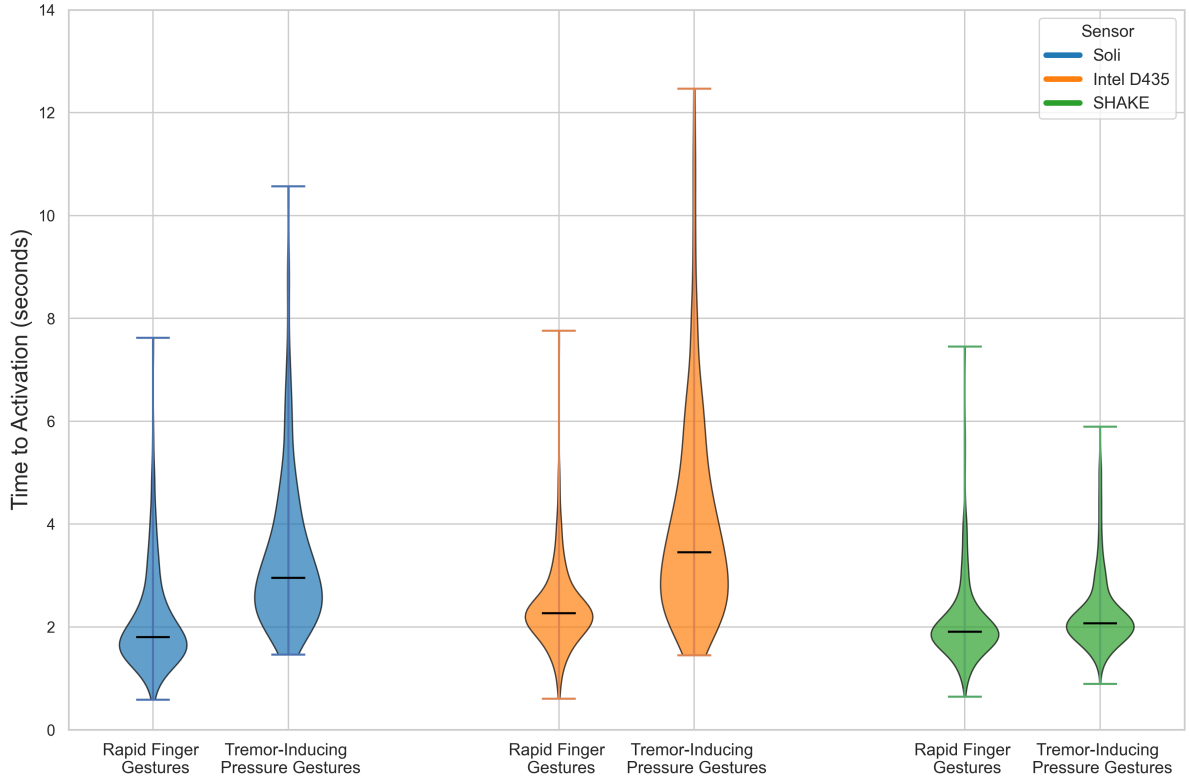


Figure 3.10: Plots illustrating the distribution of time to activation (with black line indicating the median) for *Rapid Finger Gestures* and *Tremor-Inducing Pressure Gestures* across three sensors: Soli, Mediapipe, and Sk8.

and SHAKE ($W = 0.73$, $p < 0.001$; $W = 0.79$, $p < 0.001$).

Given the non-normal distributions, the Wilcoxon Signed-Rank Test was applied to compare the time to activation between the two types of gestures for each sensor. In all cases, the p-values were exceedingly small ($p \leq 0.01$), indicating a statistically significant difference in time to activation between *Rapid Finger Gestures* and *Tremor-Inducing Pressure Gestures*: Soli ($Z = -11.96$, $p < 0.001$), Intel D435 ($Z = -11.64$, $p < 0.001$), and SHAKE ($Z = -4.39$, $p < 0.001$). Although the difference for the SHAKE sensor is statistically significant, the effect size is smaller compared to the other two sensors.

Figure 3.10 illustrates the distribution of time to activation for *Rapid Finger Gestures* and *Tremor-Inducing Pressure Gestures* across the three sensors. As seen in the plots, *Rapid Finger Gestures* generally exhibit lower median activation times and narrower distributions. This indicates that these gestures were not only faster to activate, but also more consistent

across participants, with less variability in timing compared to the broader and more variable distributions observed for *Tremor-Inducing Pressure Gestures*. Notably, while the visual separation between gesture types is most pronounced for Soli and Intel D435, the SHAKE sensor also shows this trend, albeit less distinctly—consistent with its smaller effect size despite statistical significance.

For each sensor, a composite signal was generated for each activation gesture by aligning and averaging the intensity signals across all participants at each time point. The PSD of this composite signal was then calculated. Figure 3.11 shows the PSD of each gesture across the three sensors (starting from 4 Hz for clarity). Across all sensors, *Rapid Finger Gestures* exhibit stronger high-frequency components compared to *Tremor-Inducing Pressure Gestures*. This could explain why *Rapid Finger Gestures* showed shorter and more consistent time to activation, as the sensors are better able to capture their higher-intensity, high-frequency signatures. In contrast, the subtler nature of *Tremor-Inducing Pressure Gestures* made them harder for the pipeline to recognize quickly, as these gestures do not produce high-frequency signals as strong as those generated by *Rapid Finger Gestures*. As a result, it likely took more repetitions to generate a signal strong enough to meet the recognition criteria, leading to longer and more variable activation times for *Tremor-Inducing Pressure Gestures*.

H3: *The SHAKE recognizes activation gestures faster than the other sensors.*

The Friedman test was used to determine whether the time to activation differed significantly across sensors, accounting for repeated measures within participants. Significant differences were found among the sensors for both *Rapid Finger Gestures* ($\chi^2(2) = 12.25$, $p = 0.002$) and *Tremor-Inducing Pressure Gestures* ($\chi^2(2) = 14.25$, $p < 0.001$). Post-hoc Nemenyi comparisons revealed that for Rapid Finger Gestures, both the SHAKE and Soli outperformed the Intel D435 significantly (p = 0.003 , p = 0.016 , respectively), while no significant difference was found between SHAKE and Soli (p = 0.86). For Tremor-Inducing

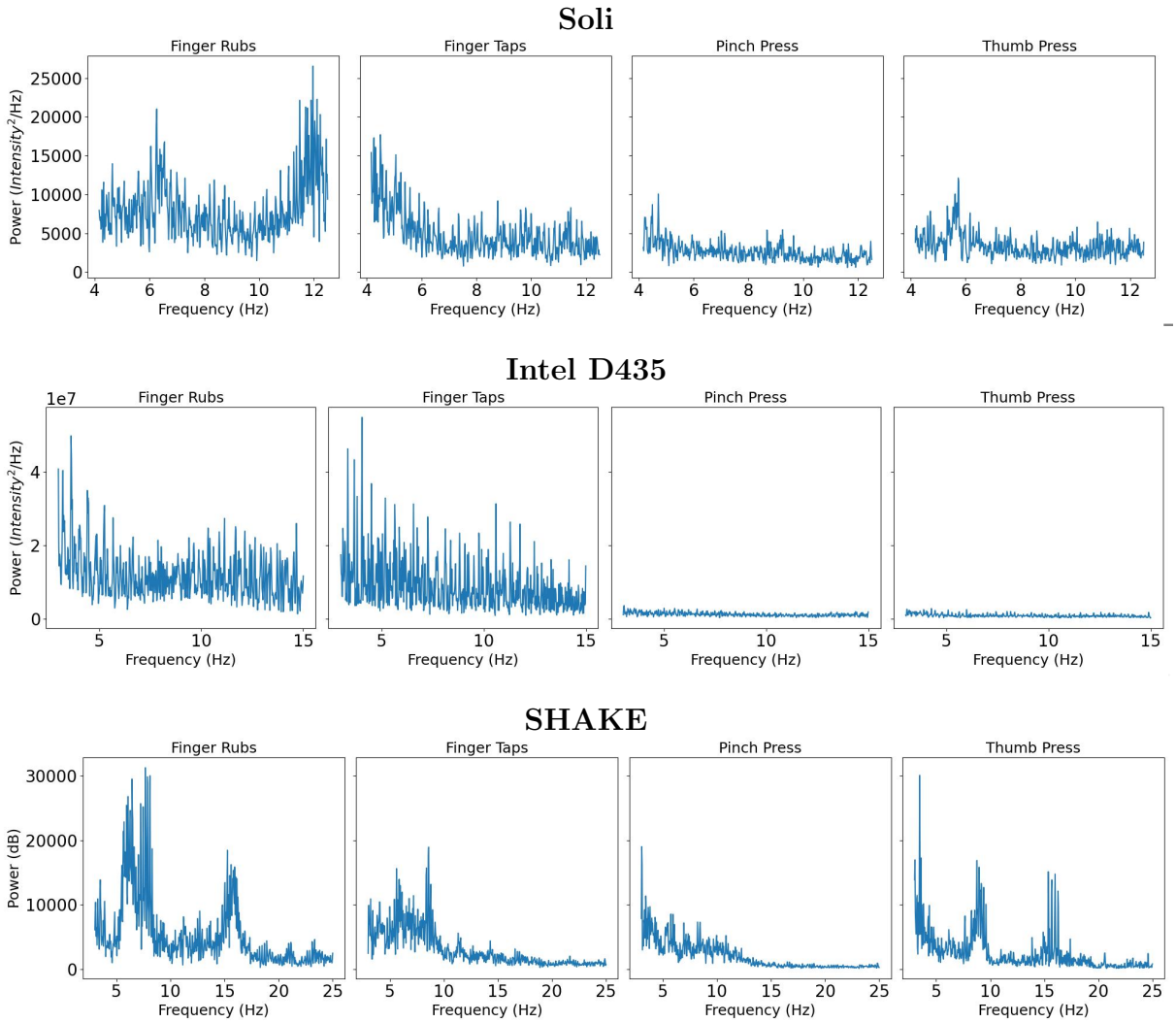


Figure 3.11: Power Spectral Density (PSD) of composite signals for each activation gesture across the three sensors (top: Soli, middle: Intel D435, bottom: SHAKE). Each plot represents the PSD of a composite signal created by averaging the intensity signals from all participants at each time point for a given gesture. *Rapid Finger Gestures* (*Finger Rubs*, *Finger Taps*) generally exhibit stronger high-frequency components across all sensors, while *Tremor-Inducing Pressure Gestures* (*Pinch Presses*, *Thumb Presses*) tend to show weaker intensity signals.

Pressure Gestures, SHAKE again outperformed Intel D435 ($p = 0.001$), but the difference between SHAKE and Soli ($p = 0.06$) and between Soli and Intel D435 ($p = 0.29$) were not statistically significant.

Figure 3.10 shows that the SHAKE has a narrower distribution for both *Rapid Finger Gestures* and *Tremor-Inducing Pressure Gestures*, with a notably lower median time to activation for *Tremor-Inducing Pressure Gestures* than the other sensors. This indicates that the pipeline recognizes gestures more consistently and quickly with the SHAKE. This can be attributed to its high sampling rate and wrist-mounted position. With a sampling rate of 60Hz, the SHAKE processes inputs at twice the frequency of the Intel D435 and more than double that of the Soli. While this enabled the SHAKE to sense gestures faster, it also made it highly sensitive to muscle movements during activities like typing, writing, and phone usage, leading to higher false activations (Table 3.2).

The Intel D435 exhibited the highest time to activation for both *Rapid Finger Gestures* and *Tremor-Inducing Pressure Gestures*, with especially high variability in the latter, as shown by the long-tailed distribution in Figure 3.10. Although the D435 has a higher sampling rate than the Soli, its vision-based tracking approach with MediaPipe struggled with the subtleties of *Tremor-Inducing Pressure Gestures*, requiring longer to recognize them.

The Soli sensor's performance in terms of time to activation fell between the SHAKE and the Intel D435. It showed better consistency than the Intel D435 in recognizing both *Rapid Finger Gestures* and *Tremor-Inducing Pressure Gestures*, but not as much as the SHAKE. However, in terms of false activations, the Soli outperformed the other sensors, recording the lowest number of false activations across all activities.

H4: *Users prefer Rapid Finger Gestures over Tremor-Inducing Pressure Gestures.*

Figure 3.12 shows the distribution of user preferences for each gesture type across the three sensors. For the Soli and Intel D435, there was a clear preference for *Rapid Finger*

Gestures, with *Finger Rubs* and *Finger Taps* being ranked in the top two preferences more often than the *Tremor-Inducing Pressure Gestures*. In contrast, with the SHAKE, *Pinch Presses* and *Thumb Presses* were more frequently ranked as the top two preferences.

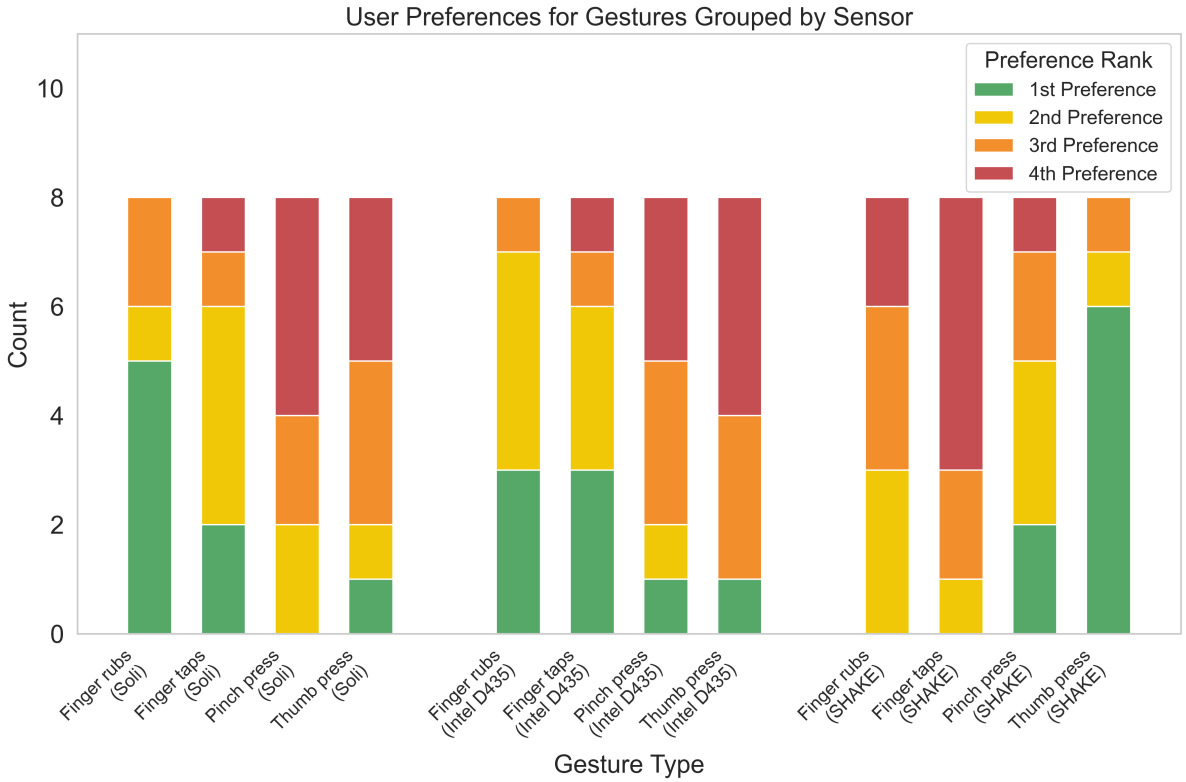


Figure 3.12: Distribution of user preferences for each activation gesture across the three sensors (Soli, Intel D435, SHAKE). Users showed a higher preference for *Rapid Finger Gestures* (*Finger Rubs* and *Finger Taps*) with the Soli and Intel D435, while *Tremor-Inducing Pressure Gestures* (*Pinch Presses* and *Thumb Presses*) were preferred with the SHAKE.

Participants primarily ranked their preferences based on how quickly they felt the gesture was recognized. The pattern of preferences aligns with the time to activation observed in Figure 3.10, where *Rapid Finger Gestures* were recognized more quickly using the Soli and Intel D435. One participant said: "The *Finger Rubs* and *Taps* were recognized faster (with the Soli and Intel D435)." This may be because these two sensors—being non-wearable and having lower sampling rates—required more repetitions to recognize the subtler *Tremor-Inducing Pressure Gestures*, making them feel slower and potentially more effortful. While these gestures appear discreet, they rely on repetitive muscle engagement

to apply force in a fixed pose. This repeated application of pressure activates deeper muscle groups in the hand and forearm, which can feel more strenuous over time compared to the lighter, more dynamic flicking motions used in *Rapid Finger Gestures*. As a result, users may have perceived the pressure gestures as more demanding despite their smaller movement range. In contrast, the SHAKE sensor demonstrated similar time to activation for both gesture types. Its wrist-mounted position placed it closer to the muscle activity involved in pressure gestures, and its higher sampling rate enabled it to capture those subtle muscle activations more effectively. This likely made the *Tremor-Inducing Pressure Gestures* feel just as responsive as *Rapid Finger Gestures* on the SHAKE. One participant noted, "I found it easier to perform the pressure gestures since it's (SHAKE) worn on the wrist."

3.6 Discussion

The proposed *activation gesture recognition pipeline* effectively ignored broad hand gestures across all sensors, demonstrating its robustness in filtering out large, low-speed, casual hand movements. However, activities like typing, writing, and phone usage produced sporadic high-frequency signals, leading to false activations. To address this, a post-hoc multi-trigger mechanism was implemented, which drastically reduced false activations across all activities for all sensors. With this improvement, **H1** is accepted.

Rapid Finger Gestures consistently achieved significantly lower time to activation than *Tremor-Inducing Pressure Gestures* across all sensors. This supports the acceptance of **H2**.

The performance of the pipeline varied notably across sensors. The Soli had fewer false activations than the SHAKE and Intel D435, while the SHAKE achieved the fastest activation times due to its higher sampling rate and wrist-mounted positioning. These results highlight how sensor characteristics significantly impact the pipeline's performance

supporting the acceptance of **H3**.

Participants' preferences for activation gestures varied across sensors. With the Soli and Intel D435, *Rapid Finger Gestures* were favored, as these gestures achieved faster recognition times and were perceived as more responsive. In contrast, with the SHAKE, *Tremor-Inducing Pressure Gestures* were preferred, due to the wrist-mounted positioning, which made these gestures feel more comfortable and comparably responsive. These variations in user preference reflect a sensor-dependent support for **H4**.

3.6.1 Limitations and Future Work

One design decision in the current pipeline was to compare total power in two frequency bands—0–4Hz and 4–12Hz—even though the bands are unequal in size. This choice was guided by empirical observations. In practice, the power in the 0–4Hz band was consistently stronger than that in the 4–12Hz band, even during high-frequency gestures. Expanding the low-frequency band would have further amplified this imbalance, making it harder to detect the subtle high-frequency gestures. Moreover, although the pipeline applies a high-pass filter to attenuate the dominance of low-frequency components, this attenuation was intentionally conservative: the goal was to suppress their overwhelming power just enough to allow high-frequency signals to emerge clearly—not to eliminate low-frequency motion altogether. Future work could explore normalization techniques to better account for differences in baseline power between bands.

The *activation gesture recognition pipeline* produced false activations during activities such as typing, writing, and phone usage, where rapid, repetitive motions occasionally generated strong high-frequency components and triggered the activation condition. To address this limitation, a post-hoc multi-trigger validation mechanism was implemented using a 2-second window with 2 and 3-trigger conditions. This modification drastically reduced false activations across all activities and sensors. However, this multi-trigger mechanism was applied post-hoc, meaning it was not tested during the real-time evaluation of the

user study, where the pipeline used a single-trigger condition. Introducing a multi-trigger condition would naturally increase the time required to recognize gestures, as multiple triggers within the specified window are needed for confirmation. While a follow-up study to evaluate this in real-time would have been ideal, it was beyond the scope of the current work which was establishing whether frequency-based activation detection could reliably distinguish intended gestures from casual hand movements. Although the multi-trigger mechanism was introduced post-hoc, the results provide meaningful insight into the effectiveness of frequency-based activation detection. The observed differences in recognition time and user preferences across gestures and sensors highlight consistent trends. These findings remain a useful foundation for gesture design and sensor integration, while future work could examine how trigger refinements affect responsiveness and usability.

There were also sensor-specific challenges with the Intel D435, where MediaPipe's hand tracking occasionally struggled with occlusion issues. When participants were performing writing and phone usage tasks, parts of their hand would sometimes be obscured by the pen or phone. These occlusions disrupted MediaPipe's ability to continuously track the hand, resulting in erratic jumps in the detected hand landmarks. This introduced high-frequency noise into the signal, which, in turn, led to some false activations. To address this challenge, a solution would involve incorporating the uncertainty in hand tracking. Currently, MediaPipe's hand tracking provides high confidence scores (typically around 99-100%) when the hand is partially occluded, which suggests an overconfidence that does not accurately reflect the quality of tracking under challenging conditions. This means that occluded hand poses, which should ideally be flagged as uncertain or low-confidence, are instead treated as reliable, leading to the introduction of noise in the signal and, ultimately, false activations. A potential solution would be to develop a probability model that could better gauge the likelihood of a valid hand pose in each frame. This would involve collecting data on valid hand poses. From there, a probability model (e.g., Gaussian Mixture Model, Kernel Density Estimation) could be used to capture the distribution of valid poses. During live gesture detection, the *activation gesture recognition pipeline* would

run alongside the model, which calculates a confidence score for each hand pose based on its similarity to valid poses. If a pose falls below a certain threshold, it could be flagged as "uncertain," which would prevent the trigger condition from being met. This approach would help the system handle occlusions and even tracking under poor lighting conditions more accurately, reducing false activations based on the reliability of hand tracking data.

Finally, because the *activation gesture recognition pipeline* was tested in a controlled environment, the next step is to evaluate it "in the wild." For example, a future user study could integrate the pipeline into a Soli-embedded smartwatch and assess its performance in real-world scenarios, such as walking outdoors, commuting on a subway, or using the watch in various public spaces. These diverse conditions would provide valuable insights into how environmental factors—like ambient noise, or erratic user movement—affect the pipeline's recognition performance and overall user experience.

3.7 Conclusion

This chapter began by addressing the challenge of recognizing activation gestures that are both subtle and distinctive, to prevent unintentional activations in sensor-based systems. Using spectral analysis, hand motions were then characterized into low- and high-frequency motions. Building on this analysis, it became evident that gestures that produce strong high-frequency components are intentional, rhythmic, and unlikely to occur by accident. This informed the selection of four candidate subtle activation gestures that would induce strong high-frequency components. An *activation gesture recognition pipeline* was then developed to ignore low-frequency hand motions and recognize the high-frequency candidate gestures. The pipeline was evaluated in a user study using three different types of sensor: mmWave radar, camera-based hand tracking, and a wearable accelerometer. Key findings from the study demonstrated that the pipeline successfully ignored broad hand motions, detecting only the intended high-frequency activation gestures. However, improvements were recommended to address instances where certain activities, such as

typing, phone usage, and writing, occasionally produced high-frequency components that could result in false activations.

3.7.1 Research Questions

The findings from this chapter contribute answers to the following research questions:

> **RQ1:** What mid-air gestures are suitable as subtle activation gestures?

The selection of suitable mid-air activation gestures involved identifying hand motions that are subtle yet capable of producing distinct high-frequency components, setting them apart from everyday movements. Through spectral analysis, gestures with high-frequency signals in the range of 4-12Hz were found to require deliberate, rhythmic repetition, making them intentional and less likely to be performed accidentally. This insight led to the proposal of four candidate activation gestures: *Finger Rubs*, *Finger Taps*, *Thumb Presses*, and *Pinch Presses*.

> **RQ2:** How can subtle activation gestures be accurately recognized without extensive data acquisition?

Accurate recognition of subtle activation gestures without extensive data acquisition was achieved by focusing on frequency-based characteristics rather than gesture-specific training data. By leveraging the unique spectral profiles of high-frequency motions, a frequency analysis-based *activation gesture recognition pipeline* was developed that detects gestures based on the power within a targeted frequency band of 4-12Hz. This approach eliminates the need for large datasets typically required by machine learning methods. User testing confirmed this method's ability in recognizing the candidate activation gestures across multiple sensor types.

3.7.2 Contributions

This chapter made the following contributions:

- Identified spectral characteristics of high-frequency rhythmic gestures and proposed four candidate subtle activation gestures.
- Developed and evaluated a versatile *activation gesture recognition pipeline* rooted in frequency analysis, that can be used with various types of sensors.

4 Subtle Gesture Recognition Using Deep Learning

4.1 Introduction

Activation is the initial command that triggers the system and sets the stage for further interaction. After activation, sensor data is continuously acquired, and the processing now shifts to recognizing specific patterns or gestures. Machine learning models, particularly deep neural networks, plays a crucial role in gesture recognition. Neural networks are exceptional at interpreting and recognizing patterns from complex data, such as range-Doppler maps (RDM). As reviewed in Section 2.5.1, these models are able to learn the temporal and spatial patterns within RDM sequences, and effectively classify different gestures. However, for neural networks to accurately recognize gestures, they require large amounts of training data.

As discussed in Section 2.5, research in radar-based gesture recognition has primarily focused on macro-gestures. The emphasis on macro-gesture recognition has been driven by several factors. These gestures are often performed using the entire arm, which generates strong radar signals which are detectable from even several meters away. Different macro gestures produce distinctive radar signatures, making it easier for neural networks to learn and differentiate between them. Consequently, neural networks can achieve high accuracy in classifying various macro-gestures due to the clear and strong patterns present in the radar data. Aside from this, macro-gestures are particularly suitable for scenarios involving private or infrequent interactions, such as turning lights on or off with a swipe

or declining an incoming call. In these scenarios, macro-gestures are useful because they are easy to perform and convenient.

With the increasing number of models and datasets focusing on macro-gesture recognition, a notable gap is emerging in the recognition of more subtle gestures. Furthermore, the absence of publicly accessible mmWave radar datasets dedicated to subtle gestures presents an opportunity for advancement in this area. This chapter aims to address this gap by introducing a new dataset of subtle hand gestures captured using the Google Soli. This dataset will be used to train various neural network models that have proven effective in recognizing macro-gestures, adapting them to subtle gesture recognition. By assessing their performance on subtle gestures, the capabilities and limitations of these models in this new context will be explored. Specifically, this chapter seeks to answer the following research question:

> **RQ3:** How accurately can neural networks recognize subtle gestures from mmWave radar data?

4.1.1 Chapter Structure

Section 4.2 outlines the rationale and design of a subtle hand gesture set. Section 4.3 describes the data collection methods, including the process of recording gesture data using the Soli. Section 4.4 presents the deep learning models used and offline experiments. Section 4.5 summarizes the main findings of the experiments, highlights the limitations, and suggests areas for future work. Finally, Section 4.6 summarizes the chapter by revisiting the research questions and outlining the contributions made.

4.2 Subtle Hand Gestures Set

Since there is a lack of prior work on radar-based subtle gesture recognition, designing an appropriate gesture set for this study presented a unique challenge. As discussed in Section 2.5.1, a review of radar-based gesture recognition literature (summarized in Table 2.1) revealed that the vast majority of prior work focuses on macro-gestures. Commonly recognized macro-gestures include swipes, pushes, pulls, and rotations, typically performed at arm scale to generate strong and easily classifiable signals. In contrast, micro-gestures have received comparatively limited attention in radar literature.

There is growing interest in micro-gestures within HCI, particularly in the context of subtle, low-effort input [56]. As defined in Section 2.6.2, the term *subtle gesture* in this thesis refers to a class of low-amplitude hand and finger motions that are spatially compact, minimally demanding in terms of physical exertion, and socially unobtrusive. This draws from prior work on micro-gestures in HCI: Chan et al. [10] described micro-gestures as "detailed gestures in a small interaction space," emphasizing miniaturization for discreet input, while Wolf et al. [92, 91] characterized them as small hand and finger movements that can be performed concurrently with another task, such as gesturing while holding a steering wheel.

Prior research—particularly in vision-based and capacitive sensing—has highlighted the value of compact, repeatable finger movements such as pinches, slides, and rotations for subtle interaction. For example, Ultraleap's micro-gesture guidelines emphasize using "only one or two fingers" in relaxed positions to perform gestures like taps, swipes, and scrubs, leveraging intrinsic proprioceptive and tactile feedback [1]. Similarly, the *HandSense* system used capacitive sensing to recognize a vocabulary of micro-gestures—including thumb-index pinches, slides, and rotational "knob-turn" motions—designed for always-available input in AR headsets [53].

In selecting the set of gestures for this research, one of the aims was to draw upon the

[1] `https://docs.ultraleap.com/xr-guidelines/Interactions/microgestures.html`

characteristics of micro-gestures from previous works such as pinches, directional slides and rotations. Another aim was to use gestures that would conform to a *gesture language*. A gesture language is a well-defined set of gestures designed to form meaningful interaction metaphors. For this research, the concept of a gesture language was realized through *virtual tools*. Complementing this, the gestures were also designed to leverage *proprioception* and *natural haptic feedback*, in order to provide intrinsic physical sensations and stability.

4.2.1 Virtual Tool Gesture Language

The concept of a virtual tool gesture language is an approach involving the use of gestures that mimic the operation of real-world objects—such as buttons, sliders, and dials. The final gesture set is illustrated in Figures 4.1 and 4.2. These gestures maintain important characteristics of macro-gestures like directionality and rotation but are adapted for subtlety, primarily relying on finger motions rather than large arm movements. For instance, in macro-gestures, swipes are typically performed using large left and right sweeping motions of the arm. In this gesture set, the swipes have been adapted into *Thumb Swipes* (Figure 4.1a) and *Pinch Swipes* (Figure 4.1b), which maintain the directionality of macro-swipes but are much more subtle. Similarly, clockwise and anticlockwise gestures, which are often performed using large circular movements of the arm or hand, have been adapted into *Index Finger Rotations* (Figure 4.2a). These gestures involve only the index finger tracing a circular path, either clockwise or counterclockwise, while the rest of the hand remains stationary. Additionally, *Close Pinch and Open Pinch* (Figure 4.1c) mimic the interaction metaphor of zooming in and out on a smartphone. Lastly, *Tick and Cross* (Figure 4.2b) also offer a clear metaphor for confirming or rejecting actions, similar to how ticks and crosses are used in written or graphical interfaces.

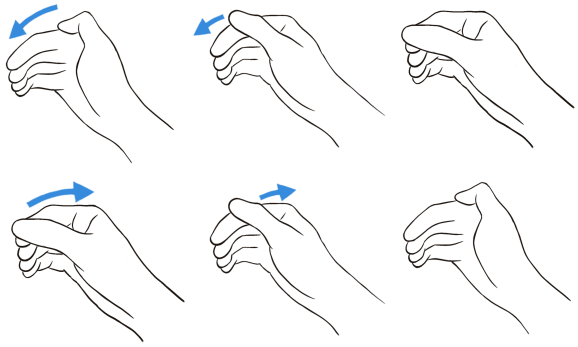
4.2.2 Haptic Feedback and Proprioception

The swipe gestures in the gesture set produce natural haptic feedback through the physical interaction of fingers. For instance, in *Thumb Swipe* and *Pinch Swipe*, the tactile sensation of the thumb sliding across the index finger or maintaining a pinch offers intrinsic haptic feedback. Additionally, the gestures were also designed to leverage proprioception, which is the body's ability to sense its position and movement in space [59]. With *Thumb Swipe* and *Pinch Swipe*, users can rely on haptic feedback to perceive the position of their fingers. Likewise, for *Close Pinch*, *Open Pinch* and *Tick* and *Cross*, users can sense their fingers' positions relative to each other by relying on internal feedback from their muscle movements and joint positions. This combination of haptic feedback and proprioception makes it easier for users to perform gestures by providing natural physical sensations and spatial awareness to guide their movements [27].

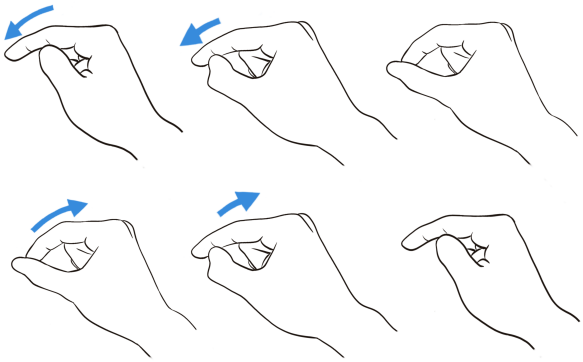
4.3 Data Collection Methods

The Google Soli radar was configured for short-range sensing in accordance with Google's recommended parameters for optimal sensing of subtle gestures. The full radar parameters can be found in Table 4.1. This configuration allowed for a maximum sensing range of 20cm and a range resolution of 2.7cm. Under this setup, the dimensions of the RDMs are 8×64, corresponding to 8 range bins and 64 velocity bins.

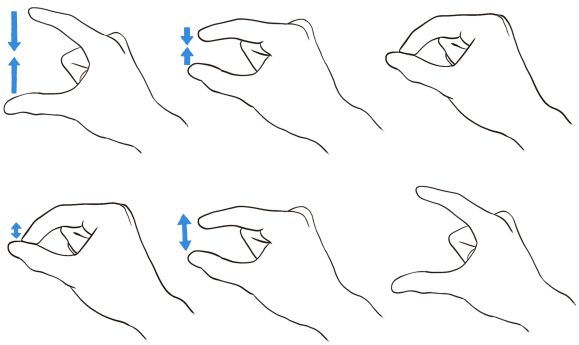
The data collection process involved capturing radar data in the form of complex RDMs as participants executed both deliberate gestures (positive data) and a variety of non-gestural motions (negative data). Central to this process was the *Constant False Alarm Rate* (CFAR) algorithm, which was used for gesture detection. The following sections explain CFAR and detail the methods used for positive and negative data acquisition.



(a) Left (top row) and right (bottom row) *Thumb Swipe*. The thumb slides across the index finger and the remaining four fingers are held parallel and steady, oriented towards the Soli.

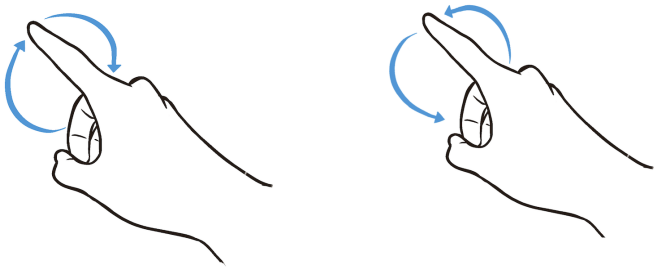


(b) Left (top row) and right (bottom row) *Pinch Swipe*. Initiated by pinching the thumb and index finger together, with the remaining fingers curled towards the palm, then sliding the thumb while maintaining the pinch, oriented towards the Soli.

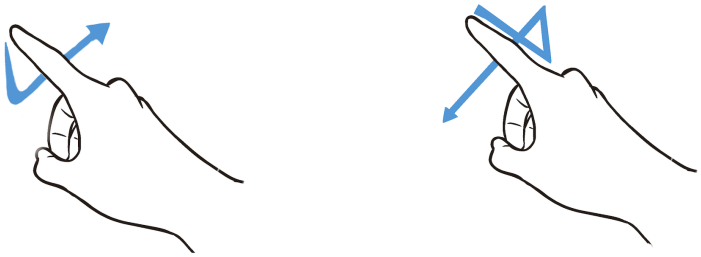


(c) *Close Pinch* (top row) with the thumb and index finger coming together from a separated position. *Open Pinch* (bottom row) starting with pinched fingers and ending with them apart. Throughout both gestures, the other three fingers remain curled towards the palm.

Figure 4.1: Subtle gesture set (1/2).



(a) Clockwise (left) and anticlockwise (right) *Index Finger Rotation*. Both gestures involve the index finger creating a circular path, while the rest of the hand remains still.



(b) *Tick* (left) made by moving the index finger in a checkmark shape. *Cross* (right) made by moving the index finger diagonally from left to right and then crosses back over in an opposite diagonal line.

Figure 4.2: Subtle gesture set (2/2).

4.3.1 Gesture Detection Using CFAR

In previous work, deep learning models have been used for continuous gesture spotting by applying a sliding window across time and classifying each segment [83]. This method can even begin classifying a gesture before it is fully complete, refining the prediction as more frames arrive. However, such continuous inference pipelines are computationally expensive and power-intensive. This becomes particularly problematic for battery-constrained devices like smartphones or wearables, where running every motion segment through a neural network would lead to excessive power consumption. To reduce this overhead for real-world deployments, a lightweight triggering mechanism can be used to first detect motion, and only then invoke the gesture recognition models.

The Constant False Alarm Rate (CFAR) algorithm plays an important role in radar systems, particularly in the context of gesture detection, by effectively managing a detection

Parameter	Value
`lower_freq`	58000 Hz
`upper_freq`	63500 Hz
`chirp_rate`	2000 Hz
`chirps_per_burst`	64
`samples_per_chirp`	16
Transmission antennas	1
Receiver antennas	3
Max sensing range	20 cm
Range bin resolution	2.7 cm
Sampling rate	25 Hz

Table 4.1: Soli parameters configuration

threshold to maintain a constant false alarm rate amidst varying noise levels [73]. The CFAR implementation in this research is derived from the methodology outlined by Choi et al. and employs an Exponential Moving Average (EMA). First, the signal intensity (absolute sum of RDM) is calculated as follows:

$$x_t = \sum_i \left\| \mathrm{RDM}^i(r, v, t-1) \right\|, \tag{4.1}$$

where RDM^i is the RDM matrix for the i-th channel. The moving average M_t at time t is then calculated using the equation:

$$M_t = (1-\alpha)M_{t-1} + \alpha x_t, \tag{4.2}$$

where $\alpha \in [0, 1]$ represents a constant smoothing factor. The gesture detection occurs if the current signal exceeds the threshold, which is defined as

$$|x_t - M_t| > \theta \cdot (M_t + M_{\text{offset}}), \tag{4.3}$$

where θ is a detection threshold, and M_{offset} is an offset parameter.

Figure 4.3 shows the operation of CFAR. The plot on the left represents a scenario where no gestures or movements are detected in front of the Soli. The absence of CFAR triggers

indicates a stable background with no motions that would suggest the presence of a gesture. The plot on the right shows red dots marking where the CFAR detection threshold is met. These red dots indicate frames when the algorithm has identified a gesture or motion that surpasses the detection threshold.

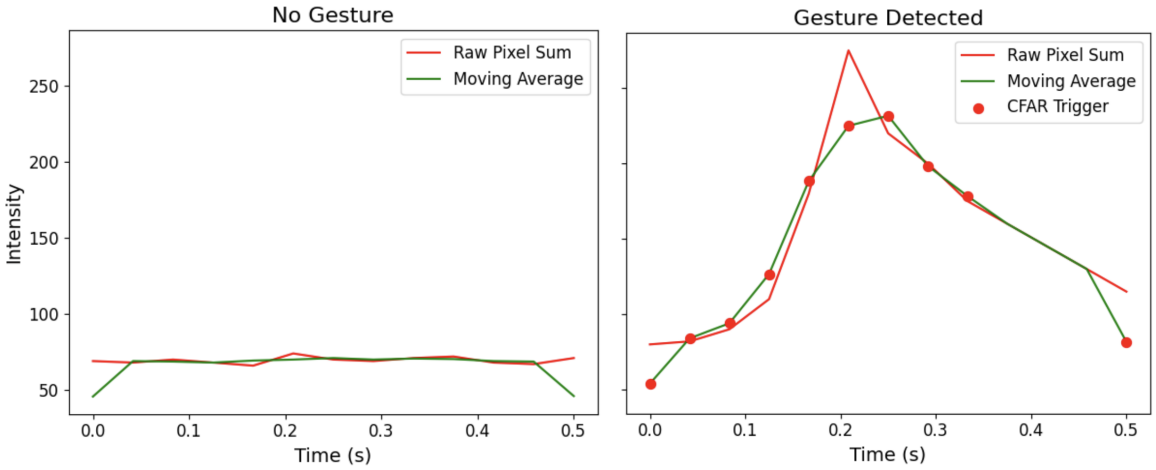


Figure 4.3: CFAR algorithm in operation. On the left, no gestures or movements are detected, resulting in no CFAR triggers. On the right, a gesture is detected, indicated by red dots where the signal intensity surpasses the detection threshold.

4.3.2 Positive Data Collection

Positive data refers to gesture instances that serve as intended input for the model to learn and recognize. Gesture data was obtained from a group of eight individuals, consisting of six males and two females, all right handed, with an age span from 24 to 36 years (average age: 27). The gestures were performed with the right hand. A total of 16,000 positive gesture samples were collected. Each sample comprises of 0.5 seconds or 13 Soli complex RDM frames. Previous research on macro-gesture recognition using mmWave radars typically employed longer windows, with the minimum being 2 seconds [14, 83, 5]. In this research, however, the rationale behind selecting half-second windows is twofold. First, each of the the subtle gestures used in this work are quicker to perform compared to macro-gestures which involve larger hand or arm movements. Macro-gestures, such as broad hand swipes, naturally take longer to execute, thereby necessitating longer windows.

Second, shorter window lengths would allow faster feedback in real-time applications. Long window lengths would introduce feedback lag due to the time taken been input and recognition.

During data collection, participants were seated comfortably in front of a laptop, with a Google Soli radar connected via USB. The Soli was positioned approximately 20cm (sensing range) from the edge of the table, and participants were instructed to perform the gestures within the edge of the table area, keeping their hands within the sensor's detection zone. Positioned behind the Soli was a camera that recorded the hand movements. A demonstrator sat to the left of the participant to oversee the process and was responsible for logging the gestures. The laptop screen displayed a live camera feed, allowing the demonstrator to monitor and verify the gestures being performed in real-time. Additionally, a live CFAR detection feed was displayed to ensure that the logged data corresponded to the intended gestures and not erroneous movements. The characteristic bell shape in the signal's moving average (right plot in Figure 4.3) represents the gesture's intensity profile, with an initial surge, peak, and decrease as the gesture is initiated, executed, and completed. This bell-shaped curve was used to identify a full gesture sample. When the CFAR detection presented this pattern, an auditory cue was emitted and the demonstrator would log the gesture by pressing the appropriate key.

The data collection session for each participant was divided into four segments, with each segment dedicated to a specific gesture type. The first segment focused on collecting data for left and right *Thumb Swipes*, the second for left and right *Pinch Swipes*, the third for *Open Pinch* and *Close Pinch*, and the fourth for *Tick* and *Cross*. Participants were given the opportunity to take short breaks between each segment or even within segments to prevent fatigue. 200 samples per gesture were collected from each participant, ensuring a substantial volume of samples for each gesture.

4.3.3 Negative Data Collection

Negative data refers to hand motions which the model should ignore. These are unintended hand motions that meet the CFAR detection criteria, such as hand adjustments or other incidental motions, that do not correspond to the defined gestures. This data was collected from the same group of participants who performed the positive gestures. The negative data collection segment was placed midway through the positive data collection session.

During the segment, participants were instructed to execute movements resembling the target gestures and any CFAR detection that did not match an actual gesture was logged as a negative sample. A key indicator for a negative sample was the absence of the characteristic bell-shaped curve in the CFAR plot. Unlike in the positive samples where a bell shape indicated correct gesture execution, any other pattern was a cue for the demonstrator to log the motion as a negative sample. Participants were also asked to move their arms towards and away from the sensor from various angles. This was done because these larger motions, compared to the subtle gestures, can trigger CFAR detection due to their strong signal intensity. Furthermore, these motions are contextually relevant as subtle gestures inherently require the hand to be brought into the sensor's detection range before the gesture can be performed, and similarly, the hand must be moved away after the interaction is completed. Finally, participants were also encouraged to perform random motions, resembling close-range adjustments. These included small, incidental hand movements such as slight repositioning of the hand or fingers near the sensor, which would trigger CFAR detection. 200 negative data samples were logged from each participant.

4.3.4 Samples Visualization

Figure 4.4 visualizes the radar data captured for the various subtle hand gestures. Each subplot represent one sample of each gesture from participant 1. The plots show the characteristic velocity changes over time as detected by the Soli. From a visual standpoint, each gesture movement corresponds to a distinctive pattern. For the left and right *Thumb*

Swipes and *Pinch Swipes*, there are vertical lines centered around the 0 velocity bins, which then break and deviate to the left or right, depending on the direction of the motion. The *Open Pinch* and *Close Pinch* also show distinct vertical lines with a short intermittent burst of intensity change highlighting the opening or closing motions of the pinch. The anticlockwise and clockwise *Index Finger Rotations* feature more dispersed patterns, reflecting the circular motion of the index finger. The *Tick* and *Cross* gestures also display dispersed patterns.

4.3.5 Data Records

A total of 17,600 gesture samples were collected comprising of 16,000 positive samples and 1,600 negative samples. The entire dataset is available for download from the *Open Science Framework* [62]. The structure of the data set is shown in Figure 4.5.

The dataset is organized into participant folders labeled 1 to 8. Within each participant's folder, there are two subfolders: *raw_data*, which contains the raw complex RDM gesture data, and *clutter_removed_data*, which contains the clutter-removed complex RDM gesture data.

The data files are stored in HDF5 format (.h5), suitable for large datasets. Each HDF5 file contains 3-channel complex RDM data corresponding to each of the three Soli receiver antenna for the specific gesture and session. The naming convention for the radar data files is pX_gesture_session_Y_type.h5, where:

- pX: represents the participant ID (e.g., p1 for participant 1).
- gesture: is the name of the gesture performed (e.g., thumbswipe, pinch).
- session_Y: indicates the session number (e.g., session_1).
- type: indicates whether the data is raw (raw) or clutter-removed (processed).

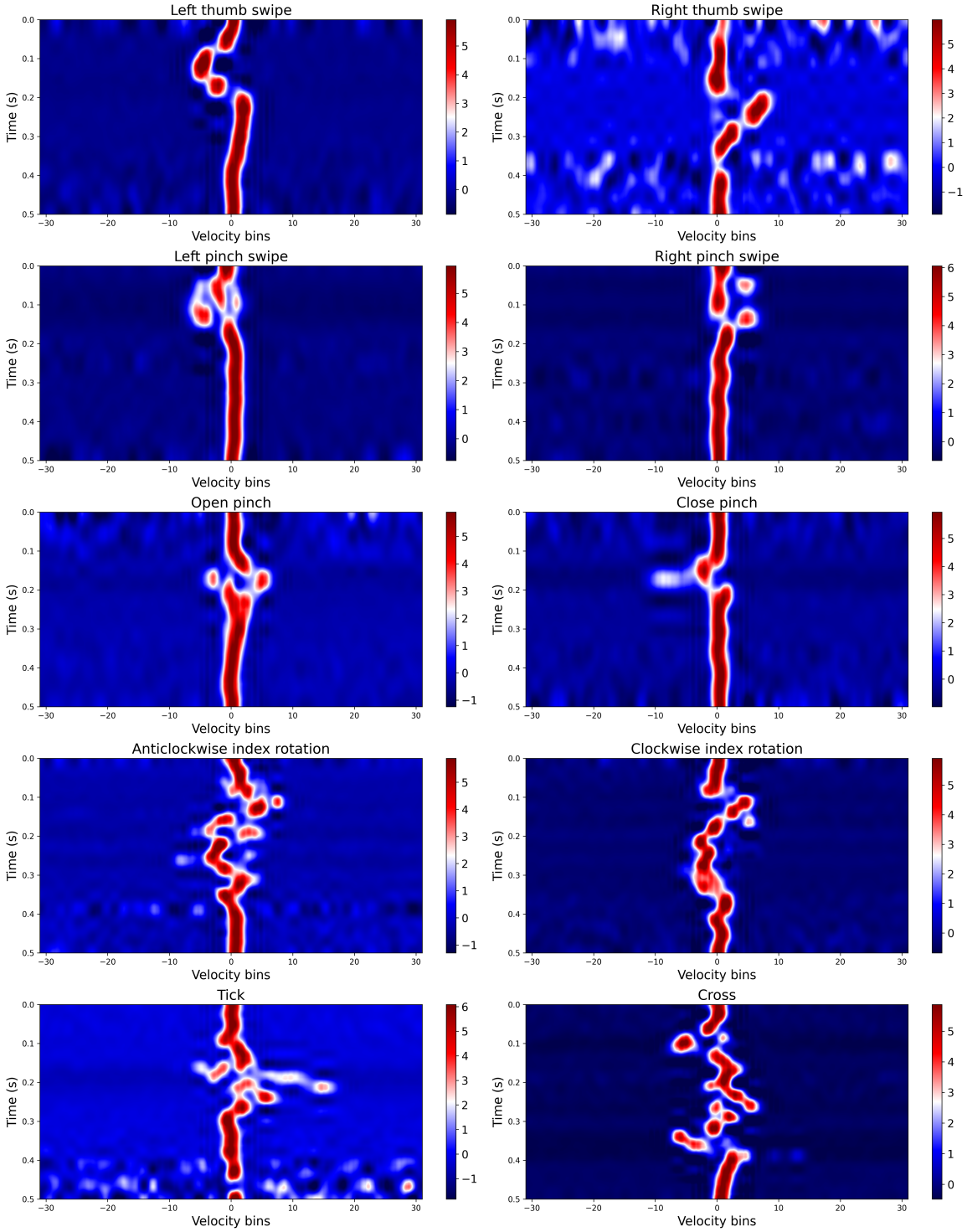


Figure 4.4: Aggregated and normalized radar responses for different subtle hand gestures captured using the Google Soli radar. Each subplot represents one sample of each gesture from participant 1, with time (0-0.5 seconds) on the y-axis and velocity bins (centered around 0) on the x-axis. The plots show the unique velocity-time signature of each gesture.

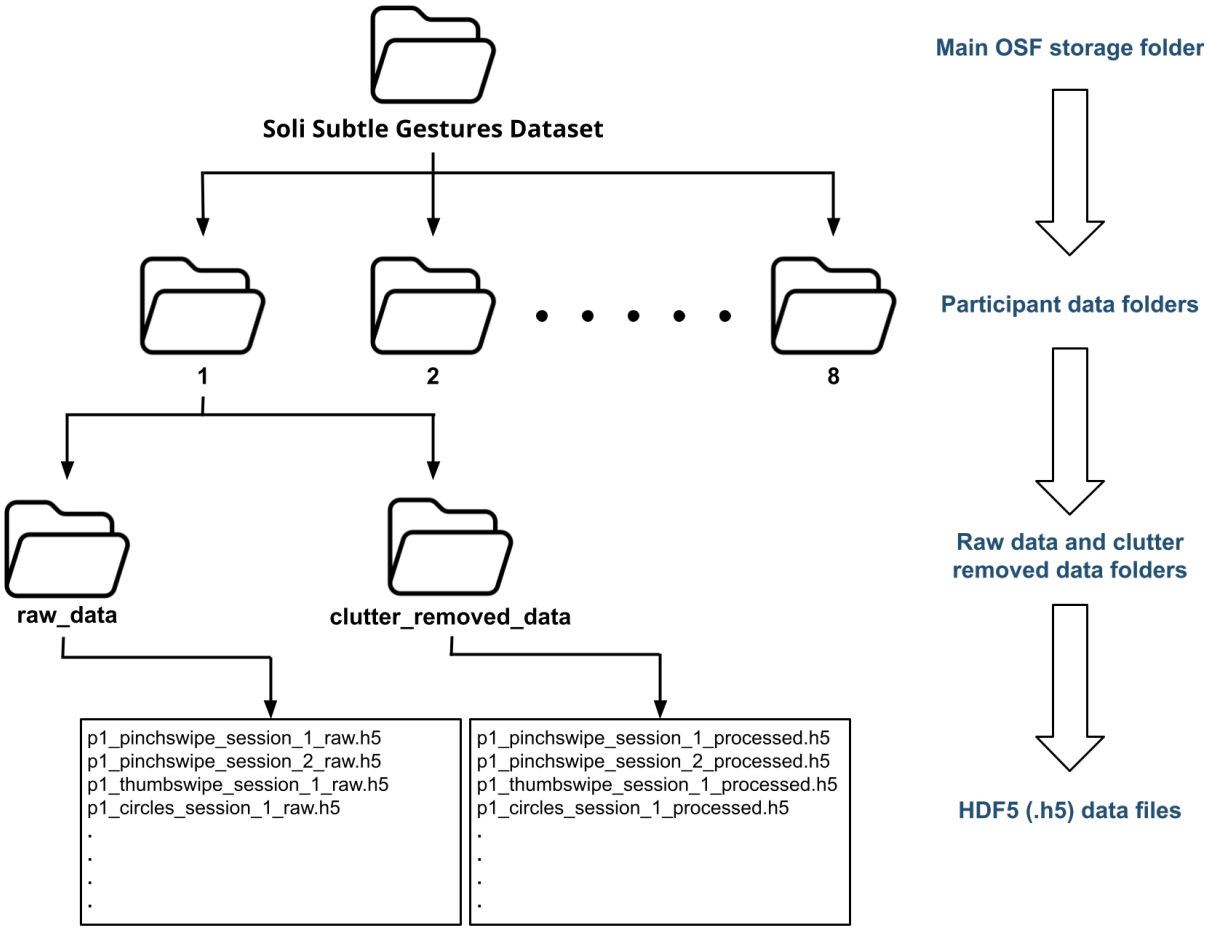


Figure 4.5: File structure of the *Soli Subtle Gestures Dataset*. The dataset is organized into participant folders, each containing subfolders for raw and clutter-removed complex RDM gesture data. Data files are stored in HDF5 format (.h5) with a naming convention indicating participant ID, gesture, session, and data type. The complete dataset is available for download from the Open Science Framework [62].

4.4 Experiments

To validate the subtle gestures dataset, various neural network architectures were trained and evaluated. The following sections describe the preprocessing steps applied to the radar data, the neural network architectures implemented for gesture recognition, and the evaluation methodology and results.

4.4.1 Data Preprocessing

The complex RDMs contain both magnitude and phase information of the radar signal returns at different ranges and Doppler shifts. In preprocessing, only the magnitude information is retained, as it captures the variations in motion patterns. The phase information is discarded due to its high sensitivity to noise, making it inconsistent and difficult to generalize across different conditions. Additionally, phase variations are often erratic and not necessarily linked to distinct gestures, whereas magnitude provides more stable motion pattern representations that are invariant to gesture-specific changes.

Following the extraction of magnitudes, the RDM from the three receiver antennas at each time step are summed together. This step aggregates the signals from all antennas, integrating multiple perspectives into a unified RDM. Finally, the aggregate RDMs are normalized.

4.4.2 Model Implementations

Three neural network architectures were implemented: a long short-term memory network (LSTM), a hybrid model combining convolutional neural network and LSTM (CNN-LSTM), and a time-distributed CNN-LSTM (TD-CNN-LSTM). CNNs are commonly used for processing spatial data such as images, where they apply convolutional filters to detect local spatial patterns like edges or textures. In the context of gesture recognition, CNNs are useful for extracting spatial features from RDMs, such as motion shapes and energy distributions across frames. LSTMs are a type of recurrent neural network (RNN) designed to model sequential data by maintaining and updating a memory of past inputs over time. Unlike standard RNNs, LSTMs incorporate gating mechanisms that allow them to retain relevant information over long sequences and discard irrelevant signals, making them especially well-suited for capturing temporal dependencies in time-series data. This property is particularly important for gesture recognition tasks, where the system must understand how motion unfolds across a series of RDM frames.

To prepare the data for input into the LSTM, each 8×64 RDM was flattened into a vector of size 512. The LSTM network processed sequences of these vectors, with each sequence comprising 13 frames corresponding to the duration of a single gesture motion. This setup would allow the LSTM to capture the temporal dependencies in the progression of the gesture across frames.

The CNN-LSTM model (Figure 4.6) and the TD-CNN-LSTM model combine the strengths of convolutional and recurrent layers. These networks process sequences of RDM data, where the convolutional layers first extract spatial features from the input data. The output of the convolutional layers is then flattened and fed into an LSTM layer, which learns the time-dependent features. This combination allows the network to first extract meaningful spatial features from each frame and then capture the temporal dependencies across sequences of frames. The key difference between the CNN-LSTM model and the TD-CNN-LSTM model lies in how they handle the sequential data. In the CNN-LSTM model, the entire sequence of frames is processed by the convolutional layers as a whole, and then the extracted features are fed into the LSTM layer. This means that the convolutional layers learn spatial features across the entire sequence simultaneously. In contrast, the TD-CNN-LSTM model applies the convolutional layers to each frame independently, and then the sequence of frame-level features is fed into the LSTM layer. This time-distributed approach ensures that the convolutional layers learn spatial features from each frame separately before the LSTM layer captures the temporal dependencies.

Training Parameters: For all models, the batch size was set to 32, and the learning rate was initialized at 0.0001. The Adam optimizer was used due to its adaptive learning rate properties. The models were trained for 250 epochs, with early stopping implemented to prevent overfitting. The early stopping criterion was set with a patience of 10 epochs, monitoring the validation accuracy.

Architecture Details: The LSTM model consisted of an LSTM layer with an input size of 512, a hidden size of 256 units, and 3 layers. Dropout was applied with a rate of 0.75 to

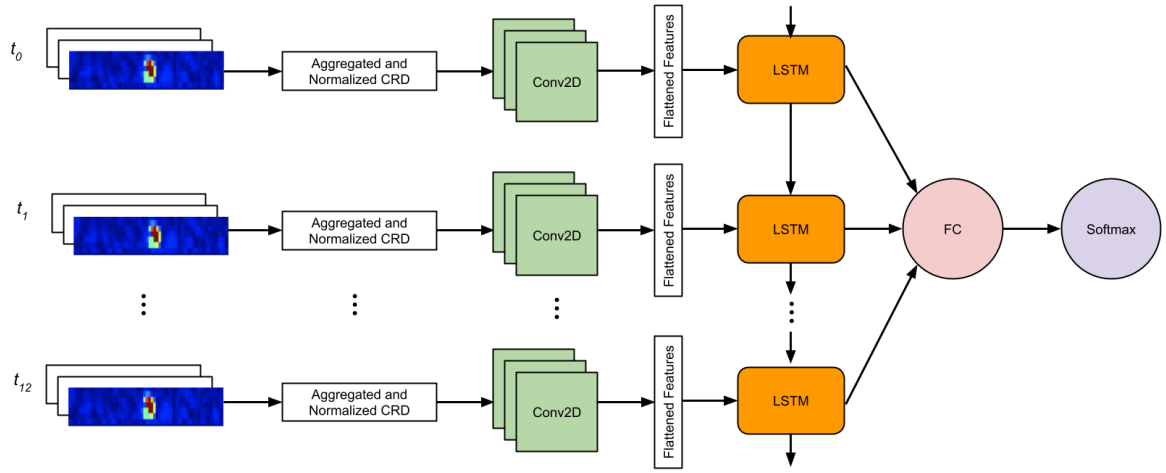


Figure 4.6: CNN-LSTM architecture used for subtle gesture recognition. Each RDM undergoes aggregation and normalization, followed by a 2D convolutional layer that extracts spatial features, which are then flattened and processed through an LSTM layer to capture temporal dependencies, culminating in a fully connected layer that feeds into a softmax layer for gesture classification.

prevent overfitting. The output of the LSTM layer was fed into a linear classifier with a hidden size of 256 units. The output of this layer was passed through a softmax function to produce the class probabilities.

The CNN-LSTM model started with three convolutional layers with 32, 64, and 128 filters, respectively, each followed by batch normalization and ReLU activation. Each convolutional layer was also followed by a max-pooling layer. The CNN part of the model included a dropout of 0.5. The output from the convolutional layers was flattened and fed into an LSTM layer with a hidden size of 256 units and 3 layers, with a dropout rate of 0.5. The output of the LSTM layer was passed through a linear classifier with a hidden size of 128 units, ReLU activation, and 0.5 dropout, followed by a final linear layer. The output of this layer was passed through a softmax function to produce the class probabilities.

The TD-CNN-LSTM model applied the convolutional layers in a time-distributed manner to each frame independently. The CNN part of the model included a convolutional layer with 32 filters followed by ReLU activation and max-pooling. The flattened output of the TD-CNN was fed into an LSTM layer with a hidden size of 64 units. The output of the

LSTM layer was passed through a linear classifier to produce the final output.

These design choices were guided by iterative tuning based on preliminary experiments on validation accuracy. For instance, the number of convolutional layers and filter sizes were selected to ensure a gradual increase in spatial abstraction. Dropout rates were empirically adjusted to balance between underfitting and overfitting, with higher dropout used in models more prone to memorizing patterns (e.g., LSTM). Hidden layer sizes were chosen to maintain enough capacity to capture gesture variations while minimizing overparameterization.

4.4.3 Results

To evaluate the models, *Leave-One-Subject-Out Cross-Validation* (LOSO-CV) was used. In this approach, data from one participant is used as the test set, while data from the remaining participants are used for training. This process is repeated such that each participant's data is used once as the test set. LOSO-CV ensures that the model is tested on completely unseen data from a new participant in each fold, providing a robust estimate of the model's ability to generalize to new users. The performance metrics, including the confusion matrix, were aggregated across all folds to provide an overall evaluation of the model's effectiveness in recognizing the gestures.

	LSTM	**CNN-LSTM**	**TD-CNN-LSTM**
Fold 1	77.1	89.5	84.8
Fold 2	79.3	90.1	85.1
Fold 3	76.5	89.8	84.9
Fold 4	80.0	90.3	85.3
Fold 5	78.9	90.2	84.9
Fold 6	79.9	89.9	85.0
Fold 7	77.6	90.5	85.2
Fold 8	78.0	90.0	85.2
Avg. (Std.)	78.5 (±1.3)	**90.0 (±0.3)**	85.1 (±0.2)

Table 4.2: Model accuracies for LSTM, CNN-LSTM, and TD-CNN-LSTM across 8 folds of leave-one-subject-out cross-validation.

The results of the LOSO-CV are summarized in Table 4.2. The LSTM model, had the

lowest average accuracy of 78.5% with a standard deviation of ±1.3. The CNN-LSTM model consistently outperformed the other models across all folds, achieving an average accuracy of 90.0% with a low standard deviation of ±0.3, indicating stable performance across different participants. This also highlights the importance of convolutional layers in extracting spatial features, which drastically improved the models perfomance compared to LSTM. The TD-CNN-LSTM achieved an average accuracy of 85.1% with a standard deviation of ±0.2, once again highlighting the advantage of integrating both convolutional and recurrent layers.

The confusion matrix in Figure 4.7 illustrates the aggregated performance of each model across 8 folds. The confusion matrix for the LSTM model (Figure 4.7a) shows the model struggling with certain gestures. For instance, right *Thumb Swipe* is frequently confused with left *Thumb Swipe* and vice versa. Similarly, the model also has a tendency to confuse clockwise and anticlockwise *Index Finger Rotations*. Right *Pinch Swipe* often gets misclassified as left *Pinch Swipe* or *Open Pinch*. This indicates that the LSTM model has difficulty in distinguishing between gestures with similar spatial characteristics. Since LSTM networks focus primarily on learning temporal dependencies in sequential data, they lack the architectural components to effectively capture fine-grained spatial patterns within individual RDM frames. As a result, they struggle to differentiate between gestures that follow similar temporal progressions but differ subtly in spatial structure — such as left versus right motions or clockwise versus anticlockwise rotations.

The CNN-LSTM model's confusion matrix (Figure 4.7b) shows improved classification accuracy across most gesture, with fewer misclassifications overall. For example, the accuracies for *Thumb Swipes, Pinch Swipes,* and *Index Finger Rotations* are much higher, indicating that the CNN layers are effectively capturing spatial features before the LSTM layers process the temporal dependencies. This improvement reflects the model's ability to capture both spatial and temporal features, reducing the confusion between similar gestures. The TD-CNN-LSTM model's confusion matrix (Figure 4.7c) also shows a better performance compared to LSTM, although slightly worse than CNN-LSTM. The lower

accuracy of the TD-CNN-LSTM compared to the CNN-LSTM could be due to the time-distributed layer overemphasizing temporal variations, which may reduce the model's ability to generalize effectively for spatially similar gestures.

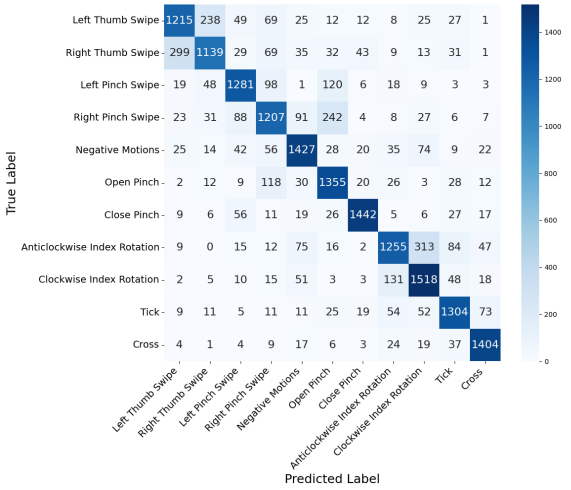
The model size and inference times are presented in Table 4.3. The LSTM model is 7.0MB and has an inference time of 1.5ms. The TD-CNN-LSTM model, at 2.3MB, is much smaller than the LSTM and also has a faster inference time of 1.3 ms. Additionally, it outperforms the LSTM in terms of accuracy, making it a more efficient choice in terms of both memory usage and speed. The CNN-LSTM model, while the largest at 13.0MB and having the highest inference time of 2.5 ms, is the best-performing model in terms of accuracy. The trade-off in model size and inference time is negligible, as the model size remains relatively small and manageable at only 13MB, making it suitable for most deployment scenarios, including edge devices and mobile platforms. Moreover, the slight difference in inference times between the CNN-LSTM and TD-CNN-LSTM models is unlikely to have a noticeable impact in real-time applications, as both have sufficiently low latencies to support smooth and responsive interactions.

Network Architecture	*Model Size [MB]*	*Inference Time [ms]*
LSTM	7.0	1.5
CNN-LSTM	13.0	2.5
TD-CNN-LSTM	2.3	1.3

Table 4.3: Resource efficiency metrics for LSTM, CNN-LSTM, and TD-CNN-LSTM models.

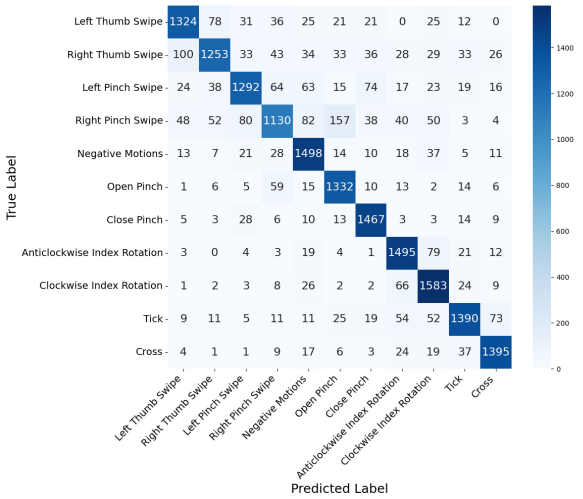
4.5 Discussion

The results of the experiments demonstrate that deep learning models can be effectively trained to classify subtle hand gestures using the collected mmWave radar data. This highlights that a sufficient volume of data has been collected for each gesture type, the data has been well-processed, and the samples of different gestures are sufficiently distinct to allow the models to differentiate between them.



(a) Aggregated confusion matrix for LSTM. The model shows difficulties in distinguishing between gestures with similar spatial characteristics, such as left and right *Thumb Swipes* or clockwise and anticlockwise *Index Finger Rotations*. The results highlight the model's limitations in capturing spatial features.

(b) Aggregated confusion matrix for CNN-LSTM. This model shows improved gesture classification accuracy, with much fewer misclassifications compared to the LSTM model. The inclusion of convolutional layers improved the model's ability to extract spatial features, leading to better recognition of gestures like *Thumb Swipes* and *Pinch Swipes*.



(c) Aggregated confusion matrix for TD-CNN-LSTM. This model shows better classification accuracy than LSTM, however, slightly lower than CNN-LSTM, possibly due to the time-distributed layer overemphasizing temporal variations.

Figure 4.7: Aggregated confusion matrices for the different neural network models over 8 folds: (a) LSTM, (b) CNN-LSTM, and (c) TD-CNN-LSTM. These matrices illustrate the performance of each model in recognizing different subtle gestures.

The CNN-LSTM model achieved the highest accuracy, consistently outperforming the other architectures with an average accuracy of 90.0%. This is attributed to the CNN's ability to capture spatial features and the LSTM's strength in capturing temporal dependencies from RDM sequences. Because of this, the CNN-LSTM model also performed well in accurately classifying gestures with similar spatial patterns. The TD-CNN-LSTM model also performed well, offering a balance between memory, efficiency, and accuracy. The performance of these models across different folds, as seen in the LOSO-CV results (Table 4.2) also highlight their ability to generalize to unseen data from different users.

90% accuracy is notable given the increased difficulty of recognizing subtle micro-gestures. In macro-gesture recognition tasks, several studies have reported higher classification accuracies exceeding 95% [19, 47]. These results are often aided by the more distinct motion signatures of macro-gestures, which involve larger arm or hand movements and produce stronger, more separable radar signals. In contrast, the gestures used in this study involve minimal finger movements, leading to weaker and more ambiguous radar signatures. Thus, achieving 90% accuracy for subtle gesture recognition using a short 0.5-second input window represents a strong result, indicating that deep learning models can successfully generalize even under more constrained signal conditions.

4.5.1 Limitations and Future Work

The radar parameters used in this study were optimized for short-range gesture sensing (Table 3.1), limiting the current models to recognizing gestures performed within a 20cm range of the Soli. Future work could focus on extending this capability to detect subtle gestures over several meters. Increasing the detection range would make the system more practical by allowing users to perform gestures from a distance without needing to approach the sensor, thereby increasing convenience and accessibility in various applications.

The Soli has one transmitting antenna and three receiving antennas. It is possible to increase the sensing range of the Soli by adjusting the radar parameters, such as lowering

the frequency range of the trasmission signal and increasing the number of samples per chirp. Although this would increase the sensing range, it would also lower the range bin resolution. Lower range bin resolution means reduced spatial precision, making it more challenging to detect subtle gestures. However, with more sophisticated radar hardware that includes additional antennas, it may be possible to detect subtle gestures from longer ranges by employing a *Multiple Input Multiple Output* (MIMO) setup. MIMO uses multiple transmitting and receiving antennas to create a diverse range of signal paths [41]. This increases the spatial resolution and allows for more precise detection of small movements over larger distances. For example, the Texas Instruments AWR6843 radar,[2] which operates in the 60-64 GHz band, offers a MIMO setup with three transmitters and four receivers and could provide fine spatial resolution for gesture detection at longer ranges. Future work could explore detection of subtle gestures from longer ranges using such radar hardware.

The dataset collected in this research, could also serve as supplementary training data for future work aiming to detect the same gestures at longer ranges. Although the radar responses at longer distances may differ due to variations in radar hardware, signal reflection, and resolution, the fundamental motion patterns encoded in the RDM data would remain consistent for the same gestures. By leveraging transfer learning, the models trained in this research, could be refined with additional long-range data. New models could also be developed by training on a combination of newly collected long-range data and the existing short-range dataset.

4.5.2 Use Cases

The *Soli Subtle Gestures Dataset* introduced in this chapter, along with the models trained on it, opens up new possibilities for low-effort and discreet interactions with technology. The gestures were selected with the virtual tools gesture language in mind, making them

[2] `https://www.ti.com/product/AWR6843`

highly practical for a wide range of applications. For instance, left and right *Thumb Swipes* are particularly well-suited for virtual sliders, making them ideal for tasks like rewinding or fast-forwarding through videos and podcasts, adjusting volume or brightness, or making discrete selections in menus or lists. *Open Pinch* and *Close Pinch* could be used to zoom in and out on digital content, mimicking common touchscreen interactions without physical contact. Clockwise and anticlockwise *Index Finger Rotations* aligns with the operation of turning a knob. *Tick* and *Cross* could be used as simple commands for accepting or rejecting inputs, similar to actions in checklists and decision-making. Each gesture offers a subtle way to navigate and control technology, potentially making mid-air interactions more accessible and efficient, especially in settings where large gestures might be impractical.

Future work could move from offline evaluation to the assessment of subtle gestures within real-time applications. The models developed in this work could be integrated into live prototypes and evaluated in controlled and 'in the wild' user studies. For example, in an automotive setting, subtle gestures could be evaluated for controlling in-car entertainment or navigation systems, helping to reduce driver distraction. In more casual scenarios, such as watching TV, subtle gestures could be used to change channels, adjust volume, or scroll through menus while sitting comfortably on a couch. Similarly, a smartwatch controlled by subtle gestures would be an excellent candidate for a public usability study focusing on social acceptability. This could investigate how people respond to using and observing subtle gestures in public to better understand the social dynamics and perceived appropriateness of subtle gesture interactions in shared spaces.

4.6 Conclusion

This chapter investigated recognition of subtle mid-air gestures using mmWave radar. While existing resources focus predominantly on macro-gestures, this research introduced a new dataset for subtle gestures. Each gesture was chosen based on the interaction

metaphors of a virtual tools gesture language and designed to leverage proprioceptive and haptic feedback. 16,000 positive data samples were collected from eight participants using a Google Soli radar, capturing 10 distinct subtle gestures. Additionally, 1,600 negative samples were collected to capture erroneous and irrelevant motions. The signal processing and data collection procedure was systematically presented, and the dataset was validated using deep learning models with the aim of answering the following research question:

> **RQ3:** How accurately can neural networks recognize subtle gestures from mmWave radar data?

Three neural network architectures were trained on the dataset using Leave-One-Subject-Out Cross-Validation, including an LSTM, a CNN-LSTM hybrid, and a Time-Distributed CNN-LSTM (TD-CNN-LSTM). The performance of each model was evaluated and presented. The final results showed that the CNN-LSTM model consistently achieved the highest performance with an average accuracy of 90.0%, followed by the TD-CNN-LSTM with 85.1%, and the LSTM model with 78.5%. The findings highlight the effectiveness of hybrid models in capturing both the spatial and temporal features of subtle hand gestures present in the RDM sequences. The results also validate the quality of the dataset, demonstrating that a sufficient volume of data has been collected for each gesture type, the data has been well-processed, and the samples of different gestures are sufficiently distinct to allow the models to differentiate between them.

This work provides a practical and well-validated framework that could inform future commercial implementations of subtle radar-based gesture control. For instance, companies like Google developing radar-driven features (e.g., Motion Sense) could use this dataset and modeling approach to expand their repertoire of subtle, socially acceptable interactions in wearables or smart environments. Additionally, the dataset and methods serve as a strong starting point for future research aimed at improving the recognition of subtle micro-gestures at longer ranges using more sophisticated radar hardware, such as MIMO-based

systems.

4.6.1 Contributions

The chapter makes the following contributions:

- Developed a new dataset specifically for subtle mid-air gestures using Google Soli radar, and made the dataset publicly available.
- Trained and evaluated multiple deep learning architectures, including LSTM, CNN-LSTM, and TD-CNN-LSTM, on the new dataset.
- Demonstrated the effectiveness of neural networks in recognizing subtle gestures, and validated the collected dataset for future use in radar-based gesture recognition systems.

5 | Exploring Slider Control Using Subtle Gestures

5.1 Introduction

Previous chapters have developed and validated systems that utilize signal processing and deep learning techniques to detect and recognize subtle gestures with mmWave radar sensors. However, the true test of any interactive system lies in its performance during live user interactions. This chapter shifts from theoretical and controlled offline evaluations to practical, real-time assessments. The work presented in this chapter explores the usability of the developed subtle gesture recognition systems in real-time applications. Additionally, this chapter examines whether subtle gestures can achieve better overall usability than macro-gestures in mid-air interaction with mmWave radar, particularly with regard to recognition accuracy, user comfort, and social acceptability.

In the scenario introduced in Section 1.1.1, Preethi wants to browse Netflix to find a new episode to watch. A virtual slider provides a suitable interaction metaphor for this kind of task, as sliding and scrolling are fundamental interactions used in navigating lists, menus, numerical entries, or timelines. Such tasks demand both directional control and stopping precision, making them a practical and representative challenge for evaluating mid-air interaction. Moreover, the neural network classifiers developed in Chapter 4 demonstrated high recognition accuracy ($\approx 90\%$) for sliding gestures, making it a natural progression to explore how those gestures perform in real-time. This chapter investigates the final research question in this thesis:

RQ4: Do subtle gestures improve task performance and user experience in radar-based interactions involving slider control?

The first step in the process of slider control is acquiring the slider handle, and the second step is moving the handle to the desired position. With a traditional cursor and mouse, this is accomplished by positioning the cursor over the slider handle and then clicking and holding the mouse button to "grab" and move the handle. For mid-air gestural interaction with sensors like Kinect or Ultraleap, users normally control a virtual cursor through arm movements in space. Various techniques have been explored to acquire the slider handle, such as moving the cursor over the handle and dwelling, using push gestures by moving the hand towards the display, as well as pinch gestures by bringing the fingers together to grasp the slider handle [87]. Once the handle is acquired, users then move it to the desired position by moving the arm left or right in space to adjust the slider's position along a predefined path.

These cursor-based interactions rely on vision-based sensors like Kinect or Ultraleap, which continuously track the user's hand movements and translate them into corresponding cursor coordinates on the screen. This enables precise control over the slider's position, much like using a physical mouse. However, this approach is not directly possible with mmWave radar, as it does not provide precise coordinate positions like vision-based systems. Instead, mmWave radar detects hand motions by measuring changes in signal reflections to determine the range and velocity of the user's hand relative to the sensor, which is embedded in the range-Doppler map (RDM). While this provides high resolution range and velocity information, it does not offer the detailed coordinate precision needed for exact cursor positioning on a screen. To allow interaction with virtual tools like sliders using mmWave radars, gesture recognition systems are required to compensate for the lack of direct coordinate tracking.

As demonstrated in Chapter 4, neural networks can leverage the spatial and temporal patterns in range-Doppler data to distinguish between even spatially similar subtle gestures

with high accuracy. Although radars do not provide explicit coordinate positions, it offers precise and consistent measurements. These capabilities can enable new forms of interaction that combine gesture recognition with radar-derived measurements—such as range, velocity, or angle of arrival—to support accurate control without relying on traditional cursor-based steering.

5.1.1 Chapter Structure

Section 5.2 describes the system design for controlling sliders using subtle gestures detected by Soli. It outlines the overall framework, including the selection of appropriate gestures, methods for gesture detection and recognition, and how these are integrated into applications for real-time interactions. Section 5.3 presents a user study which aims to assess the effectiveness of subtle gestures compared to macro-gestures and evaluate the usability of the devloped applications. Section 5.4 presents the results of the study. Section 5.5 summarizes the main findings of the evaluation, highlights the limitations and suggests areas for future work. Finally, Section 5.6 summarizes the chapter by revisiting the research questions and outlining the contributions made.

5.2 System Design

The system design overview is shown in Figure 5.1. The process begins when a user performs a gesture in front of the Soli radar. The Constant False Alarm Rate (CFAR) algorithm is then used to detect the gesture. Once a gesture is detected, the corresponding radar data is segmented and passed through a CNN-LSTM model for classification. Finally, the recognized gesture is mapped to an action within a slider-controlled application. Each of these components is explained in detail in the following sections.

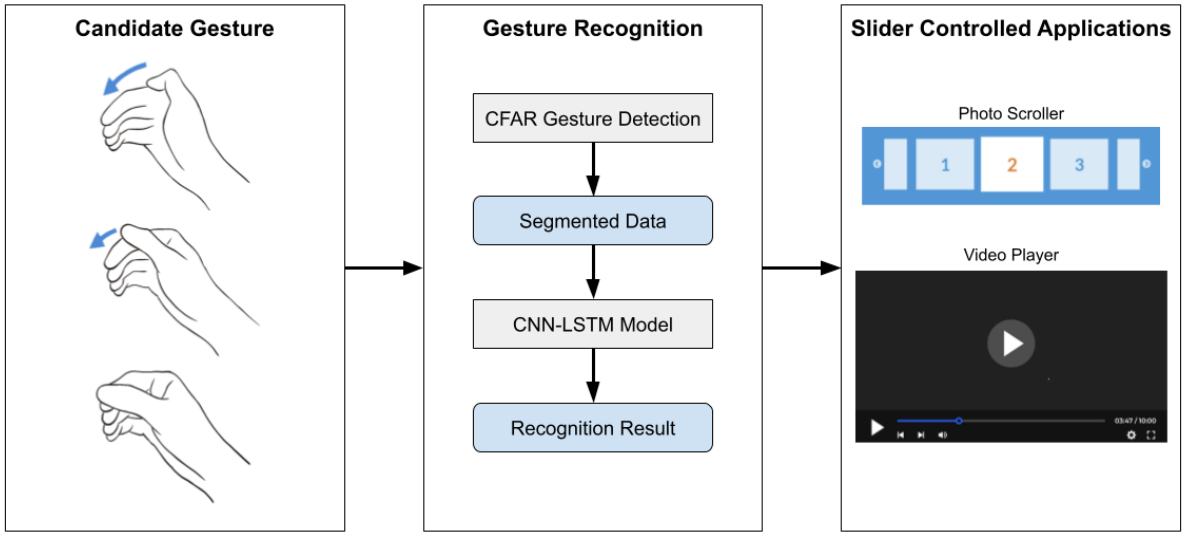


Figure 5.1: Overview of the system design for slider control using the Soli radar. Candidate gestures (*Large Swipes*, *Thumb Swipes*, *Pinch Swipes*) are performed within the Soli's detection range, detected by the CFAR algorithm, classified using a CNN-LSTM model, and then mapped to control actions in slider-based applications, such as a photo scroller or video player.

5.2.1 Candidate Gestures

The interactive systems developed for this research will focus on horizontal sliders. Given this, the candidate gestures chosen for controlling the sliders are based on directional movements. To facilitate interaction with the slider, three types of directional swipes are considered: *Large Swipes*, *Thumb Swipes*, and *Pinch Swipes*.

Large Swipes are the most common macro-gestures used for interacting with mmWave radars. This gesture involves a broad, sweeping motion of the hand either to the left or right across the radar's detection field. Due to its simplicity, ease of detection, and its popularity in gesture recognition literature with mmWave radars, *Large Swipes* serves as the baseline gesture in this study.

Thumb Swipes and *Pinch Swipes*, collectively referred to as *Subtle Swipes*, are both subtle, low-effort gestures with similar gesture profiles. *Thumb Swipes* involve the movement of the thumb to the left or right across the index finger, while the remaining fingers are kept steady and oriented toward the sensor. *Pinch Swipes* involve pinching the thumb and index

finger together and then sliding the thumb either to the left or right while maintaining the pinch. These gestures align with the interaction metaphor of virtual tools for slider control, where the index finger can be imagined as the slider itself, and the thumb moving along the index finger mimics the action of adjusting the slider handle.

5.2.2 Gesture Detection and Recognition

In real-time gesture recognition, two components work together: gesture detection and classification. For gesture detection, the system utilizes the Constant False Alarm Rate (CFAR) algorithm (explained in detail in Section 4.3.1). CFAR processes the radar data and identifies when meaningful gestures occur by analyzing the changes in radar signal patterns. Once CFAR detects a gesture, it extracts the relevant RDM segment representing that gesture.

The segment is then passed to the CNN-LSTM model for classification. The CNN-LSTM architecture is well-suited for this task because it first processes the spatial features of the RDM data through convolutional layers, capturing the structure of the gesture, and then uses LSTM layers to model the temporal dependencies across the frames, recognizing the sequence of movements that define the specific gesture. The output of this process is the gesture classification label (e.g., *Left Thumb Swipe*, *Right Thumb Swipe*, etc).

5.2.3 Applications

Once the system detects and classifies a gesture, the identified gesture is mapped to a specific action on the slider. For instance, a *Left Thumb Swipe* moves the slider handle to the left, while a *Right Thumb Swipe* moves it to the right.

Selection and seeking are two common tasks typically performed using sliders. To explore these interactions, two applications were developed: a *Photo Scroller* for *Discrete Selection* and a *Video Player* for *Continuous Seeking*.

- **Photo Scroller:** This application is illustrated in Figure 5.2a. It simulates a scenario

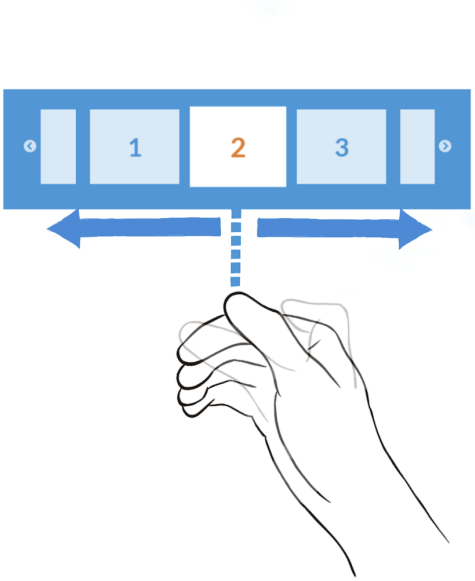
where users need to *select* fixed, discrete values from a set range, similar to scrolling through a list of photos or selecting items from a menu. The photos are arranged in a carousel, allowing users to navigate through them by swiping left or right using the specified gestures.

- **Video Player:** This application is illustrated in Figure 5.2b. It simulates a scenario where users needs to make fine adjustments along a continuous scale, such as a video timeline. The timeline is represented as a continuous slider, allowing users to *seek* forward or backward through the video by swiping left or right using the specified gestures.

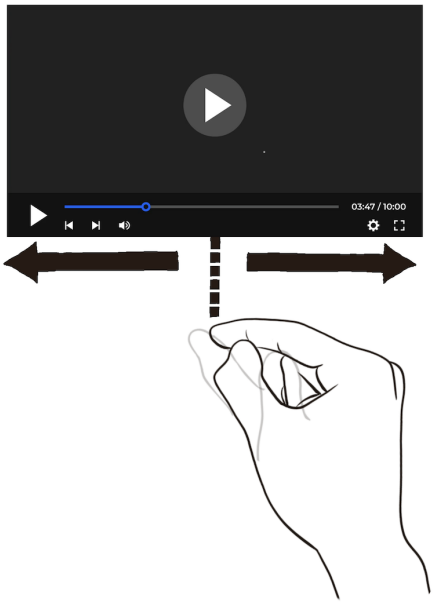
  The video player incorporates two control modes, each initiated by a discrete gesture. In the first mode, a discrete swipe gesture initiates automatic slider movement at a constant speed in the swipe direction. The movement continues until the user withdraws their hand from the radar's sensing range, which stops the slider handle. In the second mode, a discrete swipe gesture initiates slider movement, but the slider's speed is dynamically controlled by the distance of the user's hand from the radar sensor. As the user's hand moves closer to the sensor, the speed decreases toward zero; as the hand moves farther away, the speed increases linearly, reaching maximum speed at approximately 20 cm, the radar's detection limit. Further details on the signal processing steps used for deriving hand distance from radar data are provided in Appendix B.

5.3 Evaluation

A user study was carried out to determine how accurately users could perform slider-related tasks using subtle gestures. Specifically, the experiment focused on assessing the recognition accuracy of *Subtle Swipes* (*Thumb Swipes* and *Pinch Swipes*) in real-time, comparing with *Large Swipes*, and determining if users could control a virtual slider with comparable accuracy and ease. Additionally, the study sought to explore whether this



(a) *Discrete Selection Task*: Users control a *Photo Scroller* using swipe gestures to select fixed, discrete values, navigating left or right between photo items.



(b) *Continuous Seeking Task*: Users interact with a *Video Player* using swipe gestures to precisely seek through the video timeline.

Figure 5.2: Applications developed for slider-based slider control using Soli. The system enables interaction with two main applications: a *Photo Scroller* for *Discrete Selection* and a *Video Player* for *Continuous Seeking*.

type of system using radar-based subtle gestures provides a positive user experience for casual interactions. The study focuses on testing the following hypotheses:

H1: *Large Swipes are recognized more accurately than Subtle Swipes.*

This hypothesis evaluates the recognition accuracy of the neural network model in real-time performance. It is based on the expectation that *Large Swipes* generate strong radar signatures, which could lead to higher recognition accuracy compared to the two *Subtle Swipes*.

H2: *Users will complete tasks more quickly using Subtle Swipes compared to Large Swipes.*

This hypothesis tests the efficiency of the candidate gesture. It is predicated on the assump-

tion that the low physical effort required for *Subtle Swipes* will reduce task completion times.

H3: *Subtle Swipes will provide higher accuracy in controlling the slider compared to Large Swipes.*

This hypothesis evaluates the accuracy of slider control, based on the expectation that the fine motor control enabled by *Subtle Swipes* will allow users to achieve closer alignment with targets.

H4: *Users prefer Subtle Swipes over Large Swipes for slider control.*

This hypothesis is based on user experience and comfort. It assumes that the low effort and discreet nature of *Subtle Swipes* will make them more appealing than *Large Swipes*, leading to an overall user preference for *Subtle Swipes*.

5.3.1 Tasks

As mentioned before, selection and seeking are two common tasks typically performed using sliders. Therefore, two types of tasks were designed for this experiment: *Discrete Selection* and *Continuous Seeking*.

- **Discrete Selection:** This task simulates scenarios where users need to select fixed, discrete values from a set range and is implemented using the *Photo Scroller*. The photo carousel is visually set up to display numbers labeled from 1 to 10. Directly below the carousel, a target number is displayed, which the participants are instructed to navigate to using one of the canditate gestures. For example, if a *Left Thumb Swipe* is performed, the carousel moves to the number on the left, and similarly, a *Right Thumb Swipe* moves the carousel to the number on the right.
- **Continuous Seeking:** This task simulates scenarios where users need to make fine, adjustments along a continuous scale and is implemented using the *Video Player*. The application includes a progress bar at the bottom. During the task, a target

position is highlighted by a small red rectangle somewhere along the progress bar. Participants are instructed to use one of the gesture techniques to move the slider handle towards the target. Each participant is asked to complete the task using both control modes previously described: one where the slider moves at a constant speed and the other with speed control, where the slider speed dynamically adjusts based on the distance of the user's hand from the sensor.

5.3.2 Procedure

The experimental setup is shown in Figure 5.3. In the experiment, participants were first introduced to the Google Soli radar system and seated comfortably on a couch in front of a TV. A laptop connected to the radar was placed on a table in front of them, with the laptop's display projected onto the TV via an HDMI connection. This setup was designed to simulate a casual living room environment, similar to how one would watch TV at home.

Participants were then shown how to correctly perform the three candidate gestures (i.e., *Large Swipes*, *Thumb Swipes*, and *Pinch Swipes*). Specific instructions were given for *Thumb Swipes* and *Pinch Swipes* to prevent false recognition during the thumb reset motion. For example, after performing a *Right Thumb Swipe*, resetting the thumb to the left could trigger a left swipe. Although the neural networks were trained to ignore such reset movements, additional guidance was provided before the trials. Participants were instructed to reset to the starting point of the gesture by moving their thumb behind the index finger to hide it from the radar's direct line of sight. This guidance was provided to reduce the likelihood of false recognitions.

The order in which participants performed the tasks was randomized. Before beginning each task, participants were introduced to the corresponding application and allowed to interact with them. They were allowed to engage in practice trials to become comfortable with the gestures and applications.

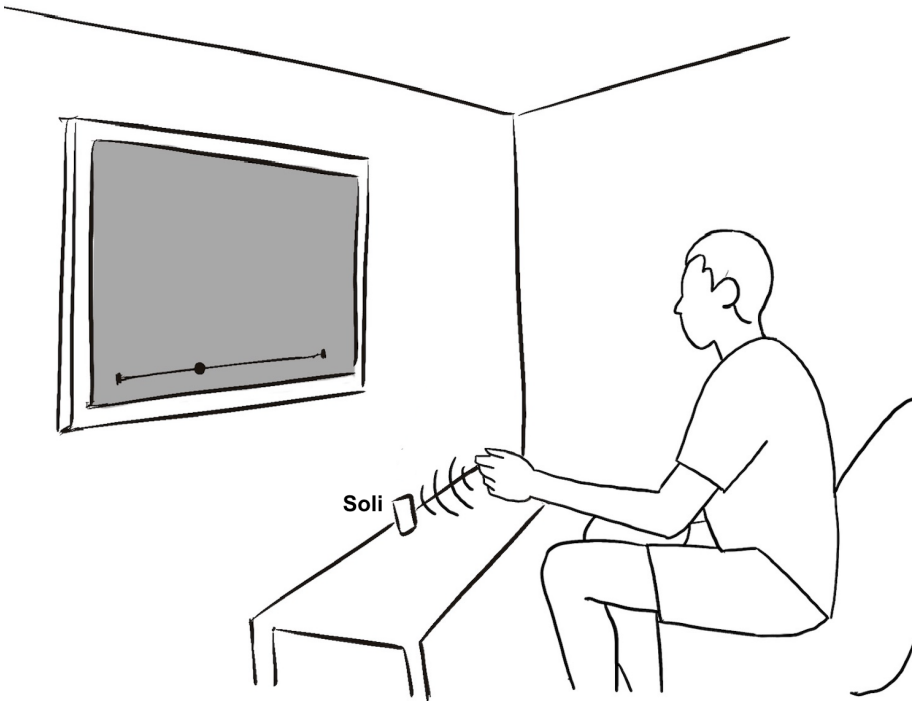


Figure 5.3: Experimental setup where participants were required to interact with the Soli radar system in a casual living room environment. Gestures performed within the detection range of the Soli sensor controlled the slider application displayed on the TV.

The order of gestures within each task was also randomized. Each session began with participants leaning forward and bringing their hands close to the Soli sensor. For *Subtle Swipes* (*Thumb Swipes* and *Pinch Swipes*), when the participant's hand came within 20 cm of the sensor, the slider would turn green, signaling that that the system is ready to recognize gestures. For all tasks, time recording started when the system recognized the first swipe gesture.

Participants were not placed under time constraints but were required to meet specific criteria for each task. For both *Discrete Selection* and *Continuous Seeking*, the objective was to hit the target 50 times, with *Continuous Seeking* requiring participants to achieve 50 target hits with both control modes. For *Continuous Seeking*, participants were also instructed to bring the slider handle as close as possible to the center of the target. The tasks ended automatically upon meeting the required criteria: for *Discrete Selection*, this was immediately upon selecting the 50th target; for *Continuous Seeking*, the task concluded as soon as the slider handle stopped within the target area after the 50th successful target

hit.

Following each task, participants were asked to complete a *User Experience Questionnaire* (UEQ) [39] to evaluate their experience with the two applications. The UEQ consists of 26 items divided into six scales, as illustrated in Figure 5.4. These include: attractiveness, efficiency, perspicuity, dependability, stimulation, and novelty. The questionnaire was also supplemented with a series of open-ended questions. These questions were designed to allow participants to explain their ratings and provide more insight into their experience with each scale. The questions were:

1. What aspects of the application did you find most appealing or unappealing? (Attractiveness)
2. Were there any moments where the application felt particularly easy or difficult to understand? (Perspicuity)
3. How did you feel about the speed and ease of completing tasks using the application? Were there any elements that made the interaction feel faster or slower for you? (Efficiency)
4. Did you feel confident that the application would consistently respond as expected? If there were any moments where it didn't, what do you think caused that? (Dependability)
5. What features of the application made the interaction exciting or boring for you? (Stimulation)
6. Did anything about the application feel new or innovative to you? (Novelty)

Additionally, participants rated their experience using a 7-point Likert scale on the following statements, which focused on *Subtle Swipes* to evaluate their usability and performance relative to the more familiar *Large Swipes*.

- **S1 Learning and Adaptation:** I was able to quickly learn and adapt to using *Subtle Swipes*.

- **S2 Goal Achievement:** I felt that I could achieve my goal faster using *Subtle Swipes* compared to *Large Swipes*.
- **S3 Precision:** I found it easier to reach the exact target with *Subtle Swipes* than with *Large Swipes*.
- **S4 Physical Comfort:** I felt physically more comfortable using *Subtle Swipes* for extended periods than *Large Swipes*.
- **S5 Public Usability:** I would feel comfortable using *Subtle Swipes* in a public setting.
- **S6 Overall Preference:** Overall, I prefer the experience of using *Subtle Swipes* over *Large Swipes* for interacting with the slider.

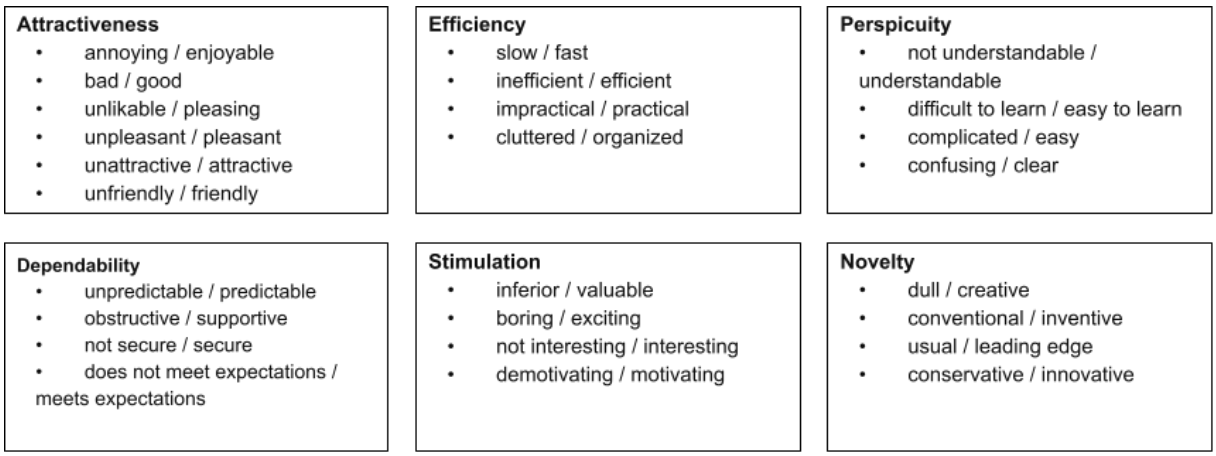


Figure 5.4: Overview of the six scales and associated items in the User Experience Questionnaire.

5.3.3 Metrics

The following metrics were used to assess the system:

- **Overall Task Time:** Measures the total duration from the initiation of the first gesture until task completion.
- **Time to Target:** Records the time it takes for a participant to move the slider from its starting position to the target.

- **Recognition Accuracy:** Measures the system's ability to correctly classify gestures in real-time.
- **Error Distance:** Measures the distance between the center of the slider handle and the center of the target.
- **Overshoots:** The number of times the slider moves past the target, requiring corrective action to move it back.
- **Undershoots:** The number of times the slider stops before reaching the target, requiring further movement to reach it.

5.3.4 Participants

Eight participants were recruited for the study, consisting of six males and two females, with ages ranging from 24 to 55 years (average age: 30). As the gesture recognition models were trained exclusively on right-hand data, all participants performed the experiment using their right hand. Ethics approval for this study was provided by the university's ethics committee.

5.4 Results

This section presents results grouped by task type. Within each task type, results are organized by the relevant performance metrics used to evaluate gesture recognition and interaction performance.

5.4.1 Discrete Selection Task

Recognition Accuracy *Recognition accuracy* for each gesture type was derived from the *Discrete Selection* task data. The accuracies are shown in Figure 5.5. The Friedman test was used to compare recognition accuracy between *Large Swipes*, *Pinch Swipes*, and *Thumb Swipes*, accounting for repeated measures within participants. No significant dif-

ferences were found ($\chi^2(2) = 1.75$, $p = 0.417$), suggesting that the CNN-LSTM model did not consistently favor one gesture type over another in terms of recognition accuracy. Although *Large Swipes* had a slightly higher mean recognition accuracy (91.3%) compared to *Pinch Swipes* (89.4%) and *Thumb Swipes* (88.2%), these differences were not statistically significant.

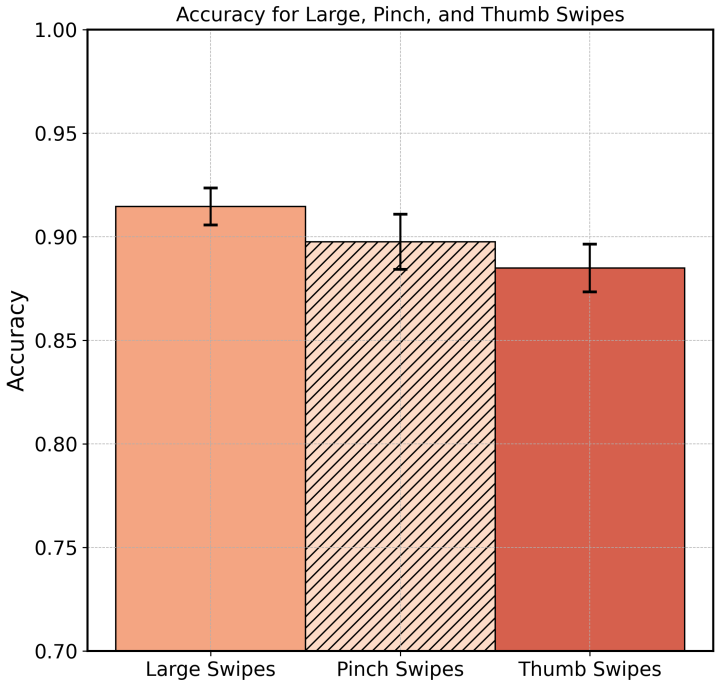


Figure 5.5: Recognition accuracy for *Large*, *Pinch*, and *Thumb Swipes* during the *Discrete Selection* task. *Large Swipes* show the highest accuracy at 91.3%, while *Pinch Swipes* and *Thumb Swipes* have slightly lower accuracies at 89.4% and 88.2%, respectively. Error bars represent the standard error of the mean (SEM) across participants.

Overall Task Time and Time to Target For the *Discrete Selection* task, the Friedman test was used to compare *task time* and *time to target* among the three gesture types (*Large Swipes*, *Pinch Swipes*, *Thumb Swipes*), accounting for repeated measures within participants. Significant differences were found for both *task completion time* ($\chi^2(2) = 12.00$, $p = 0.002$) and *time to target* ($\chi^2(2) = 14.53$, $p < 0.001$). Post-hoc Nemenyi comparisons for *task completion time* indicated that *Large Swipes* was significantly slower than both *Pinch Swipes* and *Thumb Swipes* ($p = 0.008$ for both), whereas no significant difference emerged between *Pinch Swipes* and *Thumb Swipes* ($p = 0.90$). A similar pattern was observed

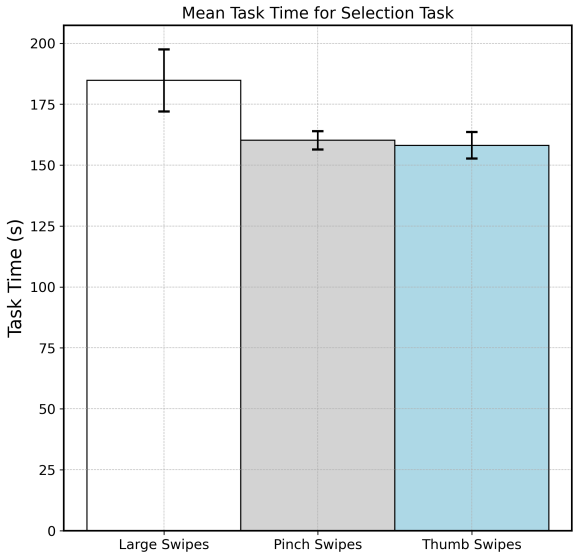
for *time to target*, where *Large Swipes* again required significantly more time than *Pinch Swipes* ($p = 0.004$) and *Thumb Swipes* ($p = 0.002$), while *Pinch Swipes* and *Thumb Swipes* did not differ ($p = 0.90$).

The mean task time (shown in Figure 5.6a) using *Large Swipes* was 184.8 seconds, compared to 160.2 seconds for *Pinch Swipes* and 158.1 seconds for *Thumb Swipes*. This represents a reduction of approximately 13% and 14% for *Pinch Swipes* and *Thumb Swipes* compared to *Large Swipes*, respectively. The mean time to target (shown in Figure 5.6b) using *Large Swipes* was 7.2 seconds, compared to 6.3 seconds for *Pinch Swipes* and 6.1 seconds for *Thumb Swipes*. This represents a reduction of approximately 13% and 15% for *Pinch Swipes* and *Thumb Swipes* compared to *Large Swipes*, respectively.

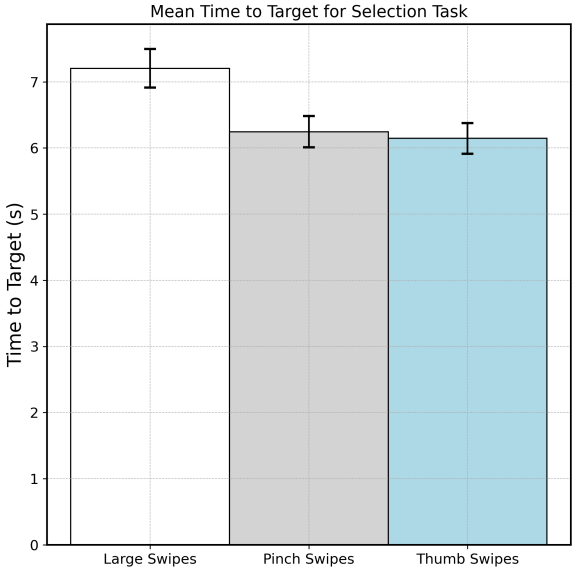
5.4.2 Continuous Seeking Task

Overall Task Time and Time to Target For the *Continuous Seeking* task, the Friedman test was first used to compare *task time* and *time to target* among *Large Swipes*, *Pinch Swipes*, and *Thumb Swipes*, accounting for repeated measures within participants. Significant differences were found for both *task completion time* ($\chi^2(2) = 9.25$, $p = 0.01$) and *time to target* ($\chi^2(2) = 14.53$, $p < 0.001$). Post-hoc Nemenyi comparisons for *task completion time* indicated that *Large Swipes* was significantly slower than both *Pinch Swipes* ($p = 0.03$) and *Thumb Swipes* ($p = 0.01$), whereas no significant difference emerged between *Pinch Swipes* and *Thumb Swipes* ($p = 0.90$). A similar pattern was observed for *time to target*, where *Large Swipes* again required significantly more time than *Pinch Swipes* ($p = 0.004$) and *Thumb Swipes* ($p = 0.002$), while *Pinch Swipes* and *Thumb Swipes* did not differ ($p = 0.90$).

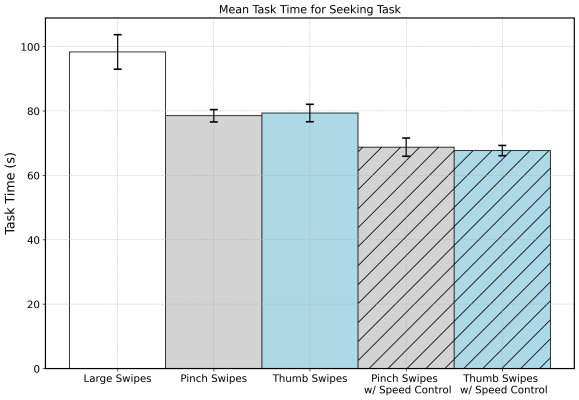
The mean task time (shown in Figure 5.6c) using *Large Swipes* was 98.4 seconds, compared to 78.6 seconds for *Pinch Swipes* and 79.4 seconds for *Thumb Swipes*. This represents a reduction of approximately 20% and 19% for *Pinch Swipes* and *Thumb Swipes* compared to *Large Swipes*, respectively. The mean time to target (shown in Figure 5.6d) using *Large*



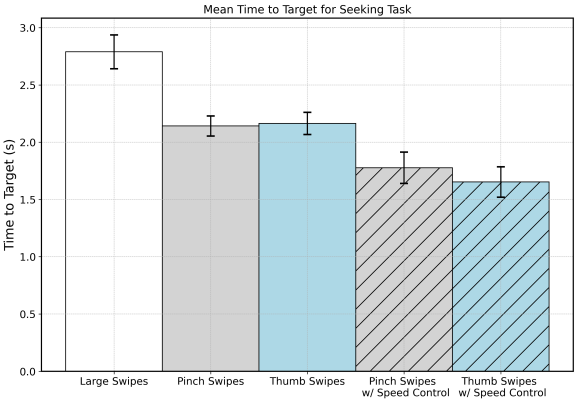
(a) Mean task completion time for the *Discrete Selection* task using *Large Swipes*, *Pinch Swipes*, and *Thumb Swipes*.



(b) Mean time to target for the *Discrete Selection* task using *Large Swipes*, *Pinch Swipes*, and *Thumb Swipes*.



(c) Mean task completion time for the *Continuous Seeking* task using *Large Swipes*, *Pinch Swipes*, and *Thumb Swipes* with and without *Speed Control*.



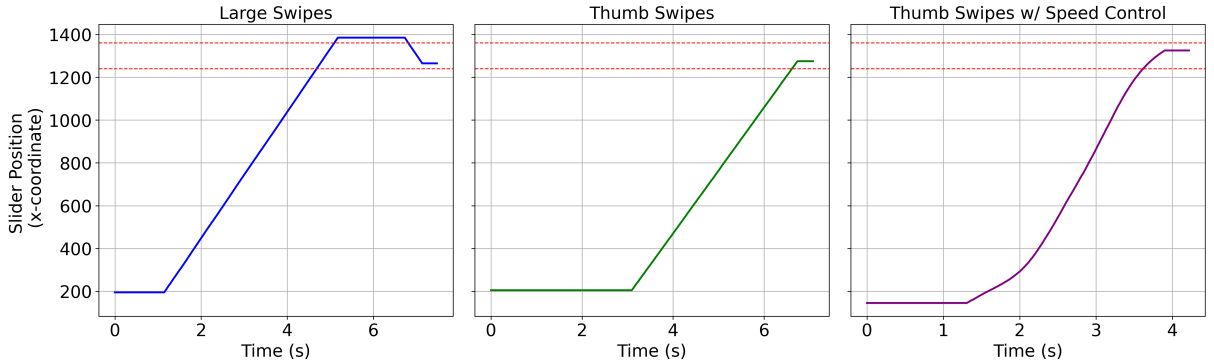
(d) Mean time to target for the *Continuous Seeking* task using *Large Swipes*, *Pinch Swipes*, and *Thumb Swipes* with and without *speed control*.

Figure 5.6: Comparison of mean task completion time (left) and time to target (right) for the *Discrete Selection* task (top row) and the *Continuous Seeking* task (bottom row).

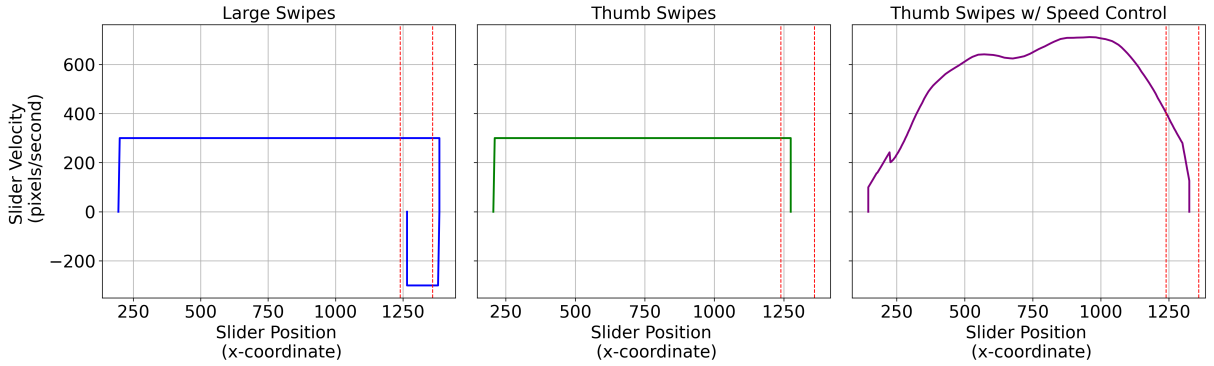
Swipes was 2.8 seconds, compared to 2.1 seconds for *Pinch Swipes* and 2.2 seconds for *Thumb Swipes*. This represents a reduction of approximately 25% and 21% for *Pinch Swipes* and *Thumb Swipes* compared to *Large Swipes*, respectively.

The introduction of speed control further reduced the mean task times to 68.8 and 67.1 seconds for *Pinch Swipes* and *Thumb Swipes*, respectively, marking approximately 30% and 32% reduction in task time compared to *Large Swipes*. The Friedman test was also used to compare *task time* and *time to target* between the *Subtle Swipes* (*Pinch Swipes*, *Thumb Swipes*) and their corresponding *speed control* versions. Significant differences were found for both *task completion time* ($\chi^2(3) = 12.15$, $p = 0.007$) and *time to target* ($\chi^2(3) = 14.67$, $p = 0.002$). Post-hoc Nemenyi comparisons for *task time* revealed that *Thumb Swipes with Speed Control* was significantly faster compared to both *Pinch Swipes* ($p = 0.019$) and *Thumb Swipes* ($p = 0.019$). For *time to target*, both *Pinch Swipes with Speed Control* and *Thumb Swipes with Speed Control* were significantly faster than their standard counterparts. *Thumb Swipes with Speed Control* was significantly faster than both *Thumb Swipes* ($p = 0.012$) and *Pinch Swipes* ($p = 0.030$), while *Pinch Swipes with Speed Control* also outperformed *Thumb Swipes* ($p = 0.047$) and *Pinch Swipes* ($p = 0.053$).

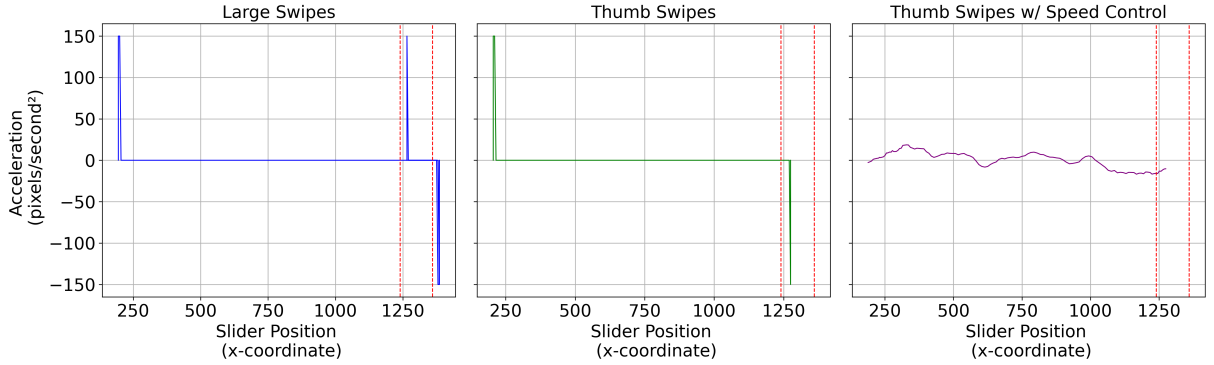
Slider Dynamics Figure 5.7 visualizes the slider dynamics for *Large Swipes*, *Thumb Swipes*, and *Thumb Swipes* with speed control (*Pinch Swipes* are omitted, as the dynamics are similar to *Thumb Swipes*). Figure 5.7a shows time series of how slider position changes over time. For all three cases, there is an initial reaction time before the slider starts moving. Then, there is a ballistic phase when the slider starts accelerating towards the target. There might then be some corrections, overshooting, and oscillations around the target (as seen in the case of *Large Swipes*). Finally, after the slider has settled on the target, it rests there for some time until the the next trial begins. Phase space plots in Figure 5.7b visualize slider velocity against its position. For *Large Swipes* and *Thumb Swipes*, the slider accelerates instantaneously (as observed in the Hooke plots in Figure 5.7c). It then moves towards the target at a constant velocity (300 pixels/second). Upon



(a) Time-series plot showing how the slider position changes over time. An initial reaction time is followed by a ballistic phase, corrections near the target, and a resting phase.



(b) Phase-space plot illustrating slider velocity over position. *Large Swipes*, *Thumb Swipes*, and *Pinch Swipes* reach a constant velocity instantly, whereas the speed control variant demonstrates a smooth bell-shaped curve, indicating gradual acceleration and deceleration.



(c) Hooke plot showing acceleration over position. These data further emphasize the instantaneous acceleration and deceleration in *Large Swipes*, *Thumb Swipes*, and *Pinch Swipes*, and the gradual acceleration and deceleration with speed control.

Figure 5.7: Slider Dynamics for various gesture types and control modes. The area between the two red dashed lines represents the target area. All trials shown are for target 7 of participant P1. The index of difficulty is 4, one of the higher IDs in the dataset.

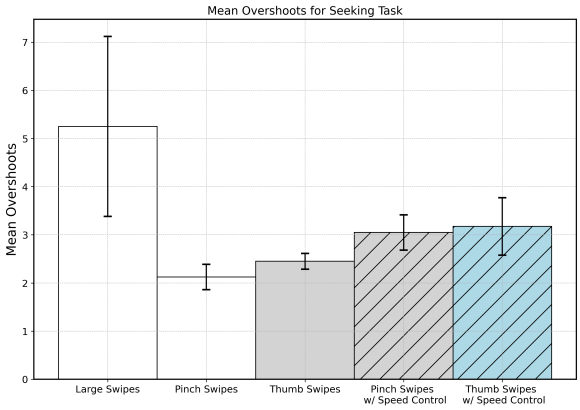
reaching the target, the slider decelerates instantaneously, bringing the velocity to zero. In these cases, users lacked the ability to accelerate or decelerate dynamically. With speed control, instead of an instantaneous acceleration to a constant velocity, there is a smooth, continuous adjustment of speed. The phase space plot for *Thumb Swipe* with speed control reveals a bell-shaped trajectory, indicating that the slider's velocity increases gradually as it moves towards the target, reaches a peak, and then decelerates smoothly as it approaches the target zone. This bell shape highlights the user's control over acceleration and deceleration, allowing the slider to reach higher speeds in the middle of its trajectory before slowing down as it nears the target.

Overshoots, Undershoots and Error Distance To evaluate accuracy in controlling the slider during the seeking task, three main metrics were used: overshoots, undershoots, and error distance.

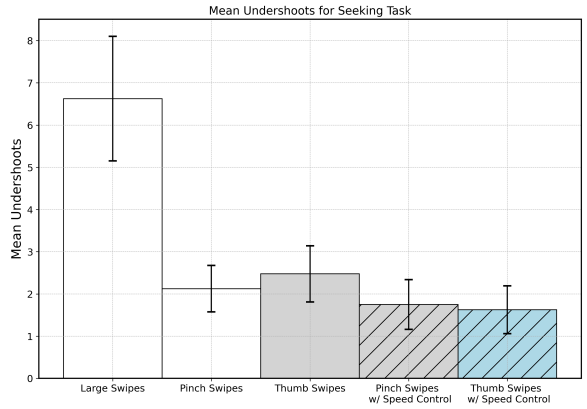
- **Overshoots:** The mean number of overshoots is shown in Figure 5.8a. Mean overshoots using *Large Swipes* were 5.2, while the mean overshoots for *Pinch Swipes* and *Thumb Swipes* were 2.1 and 2.5, respectively. This represents a significant reduction in overshoots of approximately 60% for *Pinch Swipes* and 52% for *Thumb Swipes* compared to *Large Swipes*. With speed control, the mean overshoots for *Pinch Swipes* and *Thumb Swipes* were 3.0 and 3.1, respectively. This is a significant reduction of approximately 42% for *Pinch Swipes* and 40% for *Thumb Swipes* compared to *Large Swipes*.

  To determine whether speed control significantly affected overshoot frequency, the Friedman test was applied across the four subtle swipe conditions (*Pinch Swipes*, *Thumb Swipes*, and their respective speed control versions), accounting for repeated measures within participants. The test revealed no significant differences in overshoot counts across conditions ($\chi^2(3) = 3.95$, $p = 0.27$), indicating that speed control did not significantly impact the number of overshoots.

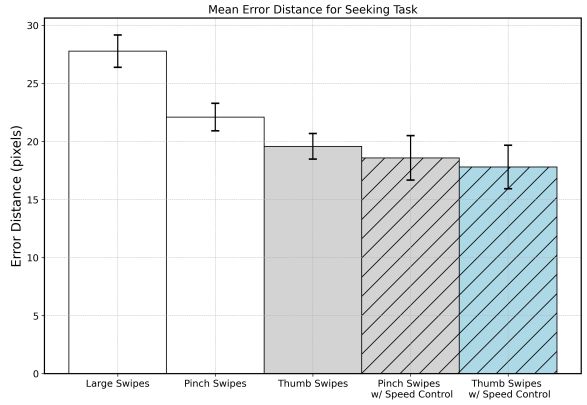
- **Undershoots:** The mean number of undershoots is shown in Figure 5.8b. Mean



(a) Mean overshoots for *Continuous Seeking* task showing significant improvement when using *Pinch* and *Thumb Swipes*, with further improvements using speed control.



(b) Mean undershoots for *Continuous Seeking* task showing significant improvement in undershoots when using *Pinch* and *Thumb Swipes*, with further improvements using speed control.



(c) Mean error distance for the *Continuous Seeking* task using *Large Swipes*, *Pinch Swipes*, and *Thumb Swipes* with and without speed control. *Subtle Swipes* with speed control exhibit the lowest error distance.

Figure 5.8: Comparison of overshoots, undershoots, and error distance for the *Continuous Seeking* task across different swipe types. Subfigures (a) and (b) present the mean number of overshoots and undershoots, respectively, while subfigure (c) shows the mean error distance.

undershoots using *Large Swipes* were 6.6, while the mean undershoots for *Pinch Swipes* and *Thumb Swipes* were 2.1 and 2.5, respectively. This represents a significant reduction in undershoots of approximately 68% for *Pinch Swipes* and 62% for *Thumb Swipes* compared to *Large Swipes*. With speed control, the mean undershoots for *Pinch Swipes* and *Thumb Swipes* were 1.8 and 1.6, respectively. This is an even more significant reduction of approximately 73% for *Pinch Swipes* and 76% for *Thumb Swipes* compared to *Large Swipes*.

To determine whether speed control significantly affected undershoot frequency, the Friedman test was applied across the four subtle swipe conditions (*Pinch Swipes*, *Thumb Swipes*, and their respective speed control versions), accounting for repeated measures within participants. The test revealed no significant differences in undershoots counts across conditions ($\chi^2(3) = 3.16$, $p = 0.37$), indicating that speed control did not significantly impact the number of overshoots.

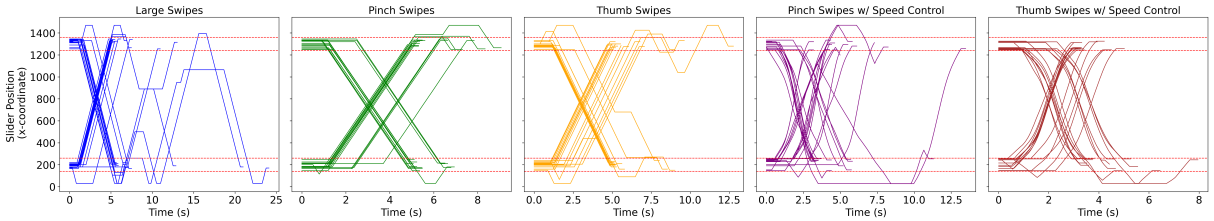
- **Error Distance:** The Wilcoxon Signed-Rank Test was used to compare error distance between *Large Swipes* and *Pinch Swipes*, and *Large Swipes* and *Thumb Swipes*. In all comparisons, $p < 0.05$, indicating a significant difference in error distance. The mean error distance is shown in Figure 5.8c. The mean error distance using *Large Swipes* was 27.8px, compared to 22.1px for *Pinch Swipes* and 19.6px for *Thumb Swipes*. This represents a reduction of approximately 20% and 29% for *Pinch Swipes* and *Thumb Swipes* compared to *Large Swipes*, respectively. With speed control, the mean error distance for *Pinch Swipes* and *Thumb Swipes* were 18.6px and 17.8px, respectively. This is an even more significant reduction of approximately 33% for *Pinch Swipes* and 36% for *Thumb Swipes* compared to *Large Swipes*.

  The Friedman test was used to compare error distance among *Pinch Swipes*, *Thumb Swipes*, and their speed control counterparts. A statistically significant difference was found ($\chi^2(3) = 9.00$, $p = 0.029$). Post-hoc Nemenyi comparisons revealed that *Thumb Swipes with Speed Control* had a significantly lower error distance than *Pinch Swipes* ($p = 0.036$). No other comparisons reached statistical significance ($p > 0.05$).

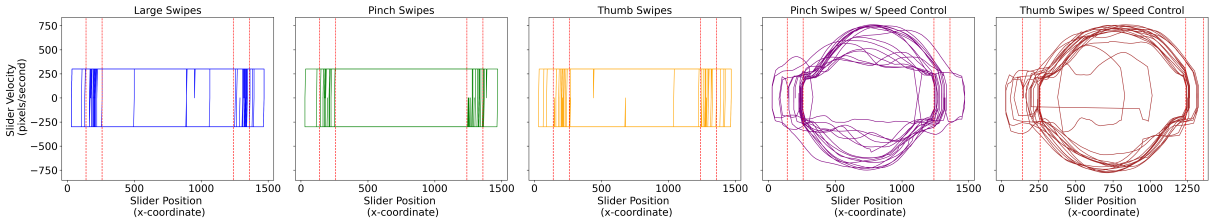
Figures 5.9a and 5.9b, display time series and phase space plots for all trials across all participants for two targets with an index of difficulty of 4. In the time series plot, the trajectories of the slider position show that *Pinch Swipes* and *Thumb Swipes*—both with and without speed control—demonstrate a more consistent approach to the targets. *Large Swipes*, in contrast, display more erratic undershooting and overshooting behaviors, as seen in the irregular paths before stabilizing on the target. Similarly, in the phase space plots for *Large Swipes* there are notable vertical lines within the plot, especially between the two target positions. These vertical trajectories indicate that the slider velocity frequently drops to zero before reaching the target, signaling frequent undershoots. In contrast, the phase space plots for *Pinch Swipes* and *Thumb Swipes* show far fewer vertical lines between the targets, reflecting less undershoots. The speed control phase space plots once again exhibit bell-shaped trajectories, as participants are able to control the slider's velocity, allowing them to reach higher speeds in the middle of the movement before decelerating as they near the target.

User Preferences For the *Discrete Selection* task, all participants preferred *Subtle Swipes* over *Large Swipes*. In the *Continuous Seeking* task, 2 of the 8 participants preferred standard *Subtle Swipes*, while the others preferred using the speed control variants. The distribution of participant responses to the six Likert statements is visualized in Figure 5.10. The key insights derived from the responses to each statement are discussed below:

- **S1 (Learning and Adaptation):** Participants generally found *Subtle Swipes* easy to learn and adapt to, with a mean rating of 5.2 (SD 0.92). Ratings ranged from 4 to 7, and no participant gave a rating below 4, indicating most users felt moderately comfortable adopting *Thumb Swipes* and *Pinch Swipes*.
- **S2 (Goal Achievement):** Participants felt they could achieve their tasks faster using *Subtle Swipes* compared to *Large Swipes*, reflected by a mean rating of 6.1 (SD 0.94). All responses were between 5 and 7.
- **S3 (Precision):** Participants reported ease in precisely aligning the slider to the



(a) Time series trajectories showing *Pinch* and *Thumb Swipes*—both with and without speed control—follow a more stable path to the target compared to *Large Swipes*, which display irregular paths with more frequent undershoots and overshoots.



(b) Phase space plots highlighting the differences in velocity control, where *Large Swipes* display frequent abrupt stops. In contrast, the speed control-enabled gestures form a smoother, bell-shaped trajectory, demonstrating better control and fewer undershoots, though occasional overshoots occur.

Figure 5.9: Slider Dynamics for *Seeking Task* for *Large Swipes*, *Pinch Swipes*, and *Thumb Swipes* (with and without Speed Control). The shown trials are for two targets with index of difficulty 4 for all participants.

target with *Subtle Swipes*, as shown by a mean rating of 5.8 (SD 0.79). Responses ranged from 5 to 7, indicating confidence in accuracy for most users.

- **S4 (Physical Comfort):** Most participants found *Subtle Swipes* physically comfortable to perform, with a mean rating of 5.9 (SD 1.05). Although responses ranged from 4 to 7, the majority indicated minimal physical strain when using these gestures.
- **S5 (Public Usability):** Participants largely indicated they would feel comfortable using *Subtle Swipes* in public, with a mean rating of 6.4 (SD 0.82).
- **S6 (Overall Preference):** Users generally preferred *Subtle Swipes* over *Large Swipes*, as evidenced by a mean rating of 6.1 (SD 0.94). All responses fell between 5 and 7, indicating a strong overall inclination toward subtle gestures.

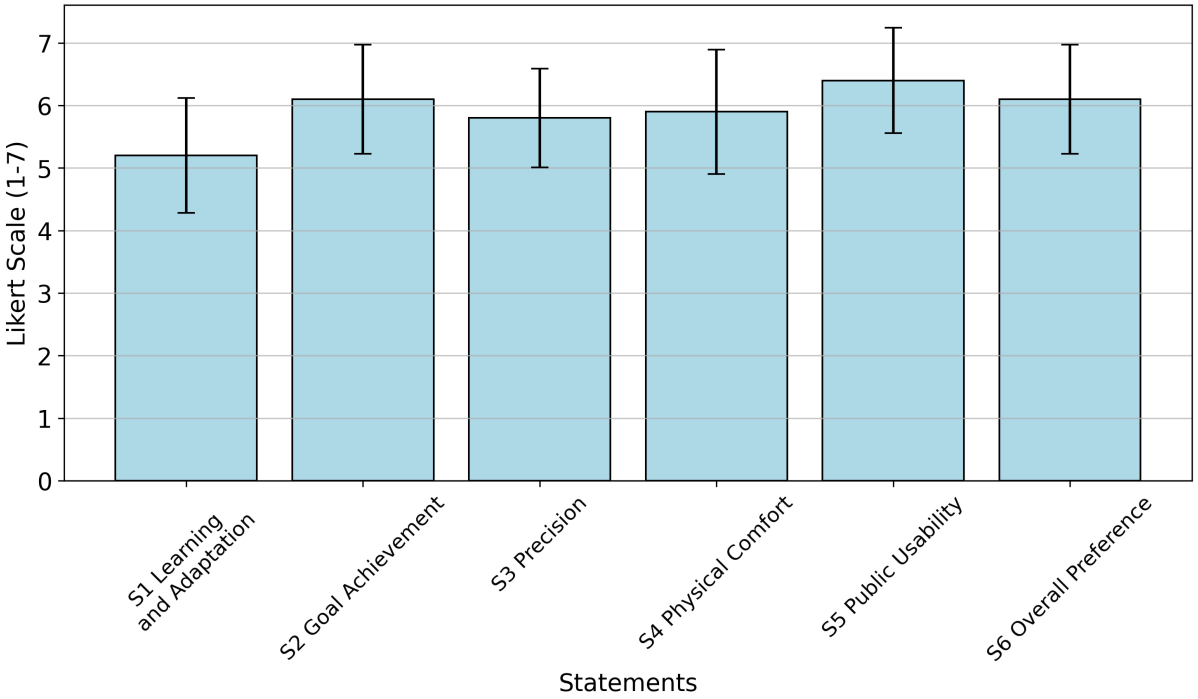


Figure 5.10: Mean Likert-scale ratings (1–7) for each of the six statements, with error bars indicating the standard deviation. Higher ratings reflect stronger agreement. Participants generally rated *Subtle Swipes* favorably, finding them easy to learn, efficient, precise, comfortable, publicly usable, and overall preferable compared to *Large Swipes*.

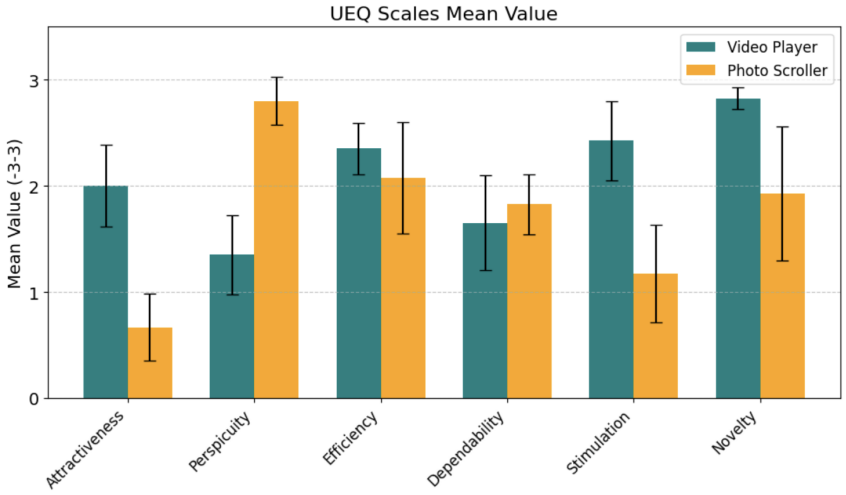
5.4.3 UEQ Results

Figure 5.11a shows the mean scale scores for the *Photo Scroller* and *Video Player*. The mean scale scores range from -3 (horribly bad) to 3 (extremely good) [39]. The scores show that both applications performed positively across all scales. Figure 5.11b further break down the results by showing the mean value per item across each of the six scales. The following analysis offers a closer look at each scale, revealing participant perceptions and highlighting strengths and areas for improvement with both applications.

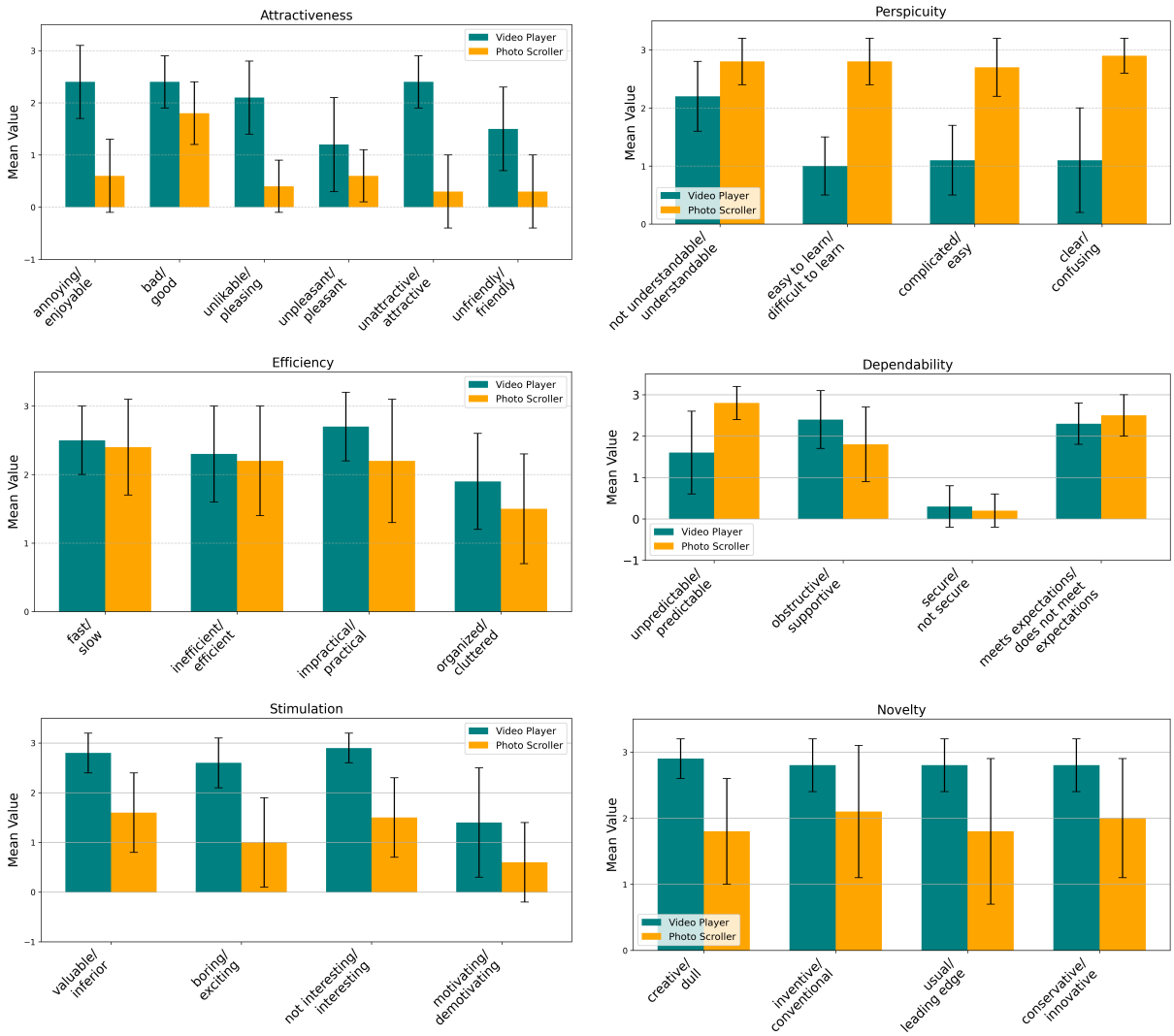
- **Attractiveness**: This scale evaluates the overall appeal of the application, capturing users' impressions of its visual design, aesthetics, and how pleasant they found the interaction. The *Video Player* showed a mean of 2.0 (SD 0.39), while the *Photo Scroller* had a mean of 0.67 (SD 0.32). Looking at the itemized results, the *Video Player* outperformed the *Photo Scroller* across all attributes. The responses from the open-ended questions indicated that participants found the *Video Player* more

familiar. One participant commented, "It (*Video Player*) had a layout and design similar to what I use daily, which made it easier to connect with." In contrast, multiple participants noted that the *Photo Scroller* felt unfamiliar: "Its (*Photo Scroller*) design wasn't like what I'm used to on my phone or iPad."

- **Perspicuity**: This scale assesses how easily users could understand and become familiar with the application. The *Photo Scroller* received a higher mean score of 2.80 (SD 0.22), compared to the *Video Player's* 1.35 (SD 0.37). The itemized breakdown indicates that participants found the *Photo Scroller* easier to learn and generally less complicated. One participant stated, "It was immediately clear how to scroll through the images." In contrast, the *Video Player* required more time to learn: "It took me longer to understand how the hand gestures worked with the speed control." The *clear/confusing* item shows the *Photo Scroller* being rated as clearer. One participant mentioned that the *Video Player* controls were more complex: "At first, it wasn't obvious how to control the video using gestures, and I had to try a few times to get it right."
- **Efficiency**: This scale evaluates whether users can accomplish their tasks without unnecessary effort. The *Video Player* had a mean of 2.35 (SD 0.25), and the *Photo Scroller* 2.08 (SD 0.53). One participant noted regarding the *Video Player*, "I liked how controlling the speed made seeking faster." The itemized results show that participants appreciated the practical aspects of both systems, with both applications scoring high for usability and reduced effort.
- **Dependability**: This scale measures user control, focusing on how reliable and predictable the system feels. The *Video Player* scored a mean of 1.65 (SD 0.45), and the *Photo Scroller* 1.83 (SD 0.28). Some participants rated the *Photo Scroller* as more *predictable*, with one noting, "The *Photo Scroller* responded to every swipe consistently, so I felt like I was in control the whole time." Another participant mentioned that the *Video Player* initially felt less predictable, explaining, "At first, the speed control was tricky to manage, and it felt unpredictable. But after a few



(a) Comparison of mean scale scores between the Video Player and Photo Scroller applications.



(b) Mean value per item for all UEQ scales (Attractiveness, Perspicuity, Efficiency, Dependability, Stimulation, and Novelty) for Video Player and Photo Scroller applications.

Figure 5.11: (Top) Overview of mean scale scores; (Bottom) Mean value per item across six UEQ scales.

tries, it became easier."

- **Stimulation**: This scale examines whether using the system is exciting and motivating. The *Video Player* received a higher mean score of 2.43 (SD 0.37), whereas the *Photo Scroller* scored 1.18 (SD 0.46). Participants found the video interactions more engaging, with one stating, "The ability to control the speed while seeking through the video made it more interesting." The *Photo Scroller* was seen as functional but relatively less stimulating.
- **Novelty**: This scale focuses on how innovative or unique the system appears. The *Video Player* achieved a mean of 2.83 (SD 0.10), while the *Photo Scroller* scored 1.93 (SD 0.63). Many participants highlighted the novelty of radar-based interaction, with one remarking, "The radar aspect was new to me. I've never controlled a video player without touch before, and the experience felt futuristic."

5.5 Discussion

Despite the expectation that *Large Swipes* would be recognized more accurately than *Subtle Swipes*, the results showed no significant difference in recognition accuracy between the gestures. *Large Swipes* had a slightly higher mean recognition accuracy (91.3%) compared to *Pinch Swipes* (89.4%) and *Thumb Swipes* (88.2%), but these differences were not statistically significant, leading to the rejection of **H1**.

Pinch Swipes and *Thumb Swipes* reduced task time by up to 14% in the *Discrete Selection*. The faster performance can be primarily attributed to the nature of the movements. *Large Swipes* require the user to move their whole arm, necessitating significant motion and time to reset each swipe. In contrast, *Pinch Swipes* and *Thumb Swipes* keep the hand positioned in front of the Soli without needing to reposition the entire arm. Only minimal finger movements are involved, allowing for faster gesture repetitions. In the *Continuous Seeking* task, *Pinch Swipes* and *Thumb Swipes* reduced task time by up to 14% . When combined with *speed control*, seeking task times were further reduced by up to 32% compared to

Large Swipes. *Speed control* enabled finer control over slider velocity by allowing users to gradually accelerate and decelerate through small hand movements toward and away from the Soli. This continuous velocity adjustment allowed users to reach higher peak slider speeds while maintaining precision near the target, thereby reducing both task time and time to target. These reductions highlight the efficiency gains provided by *Subtle Swipes*, supporting the acceptance of **H2**.

The number of overshoots and undershoots reduced by up to 60% and 68%, respectively, when using *Subtle Swipes* instead of *Large Swipes* in the *Continuous Seeking* task. With *speed control*, undershoots were further reduced by up to 76% compared to *Large Swipes*. The addition of *speed control* enabled users to make more deliberate and fine-grained adjustments near the target, which significantly reduced the number of undershoots. However, overshoots increased in some cases (although not statistically significant), resulting from participants not decelerating early enough before reaching the target. Error distances reduced by up to 29% with *Subtle Swipes* and up to 36% with *speed control*. The ability to control velocity continuously allowed participants to slow down as they approached the target, helping them converge more closely to the target center and reducing error distance. While the absolute differences in error distance ($\approx$ 8–10 pixels) may appear small relative to the total slider range, even these differences can be meaningful depending on the application. For instance, in long-form content such as movies or surveillance footage, each pixel on the slider might correspond to several seconds or minutes of video. In such contexts, smaller error distances directly translate to more accurate temporal navigation. Moreover, these gains were accompanied by substantial reductions in overshoots and undershoots, reflecting more stable and controlled adjustments near the target. These metrics indicate that *Subtle Swipes* provide higher accuracy in controlling the slider compared to *Large Swipes*, supporting the acceptance of **H3**.

Finally, participants consistently favored *Subtle Swipes* over *Large Swipes* in terms of goal achievement, precision, comfort, and social acceptability, as reflected by the Likert scale responses (Figure 5.10). For both the *Photo Scroller* and *Video Player* applications, all

participants preferred using *Subtle Swipes* over *Large Swipes*. While *speed control* was generally well received, a few participants noted an initial learning curve, describing it as tricky to manage at first. After a few trials, however, the interaction became easier to handle and felt more predictable, highlighting how practice helped mitigate early uncertainty. Overall, the results support the acceptance of **H4**.

5.5.1 Limitations and Future Work

The CNN-LSTM model used in this chapter was trained on radar data limited to a 20cm detection range. As a result of this, the applications designed in the user study required participants to perform gestures within this range of the Soli. Predicated on the possibility of extending the detection range of subtle gestures (discussed in Section 4.5.1), future work could explore designing applications for long-range subtle gesture interaction. This could focus on evaluating the usability of subtle gestures in scenarios where users can interact with devices from a few meters away, such as controlling a TV or smart speaker from across a room.

Another limitation of this work is the lack of 'in the wild' evaluation to assess social acceptability and ethical issues with subtle interactions. While the subtle gestures employed in this research are inherently designed to be discreet and low-effort, the social acceptability of these interactions were only assessed through self-reported measures like Likert scale statements (e.g., "I would feel comfortable using subtle swipes in a public setting"). A more holistic evaluation would involve user studies in public or semi-public spaces to understand how observers perceive subtle gestures when participants engage in interactions with devices, capturing both user comfort and social acceptance from a bystander perspective.

One of the benefits of subtle interactions is that they do not draw attention to the user. However, this also presents a double-edged issue — what is subtle from the user's perspective may be deceptive or "sneaky" for the observer. For instance, imagine a family watching TV together when, suddenly, the channel is changed without anyone knowing

who issued the command. Similarly, consider at a dinner where one person is secretly interacting with their device, deliberately concealing it from their date. The use of subtle gestures could obscure accountability and lead to misunderstandings or even suspicion among observers, especially when they are not aware of the gesture-based control.

The discreet nature of subtle interactions that make them socially acceptable for the user can simultaneously obscure their actions from others, creating a tension between the desire for discreet control and the need for observable, accountable behavior. The implications of this duality must be considered, especially in scenarios where shared or public contexts are involved. Future work could explore how to balance discreet interaction with transparency for observers by focusing on feedback that would help observers remain aware of ongoing interactions without disrupting the subtle nature of the gesture itself. For example, in the case of the *Video Player*, feedback mechanism could involve visual cues on the screen showing the direction from which the subtle gesture was performed, allowing observers to locate who made the change. Other modes like light feedback could also be effective. For instance, a smart speaker could employ a ring of LEDs that briefly light up in the direction from which the gesture was performed.

5.6 Conclusion

This chapter explored radar-based slider control using subtle gestures. First, a framework was presented to control slider-based applications using a set of three candidate gestures (*Large Swipes*, *Pinch Swipes*, and *Thumb Swipes*). Two applications were designed and integrated to enable slider control using a Google Soli radar: a *Photo Scroller* and a *Video Player*. A user study was then carried out to answer the following research question:

> **RQ4:** Do subtle gestures improve task performance and user experience in radar-based interactions involving slider control?

The user study evaluated the effectiveness of *Subtle Swipes*(*Pinch Swipes*, and *Thumb Swipes*) compared to *Large Swipes*. Eight participants were recruited to perform two types of tasks with the developed applications. The *Photo Scroller* was used for simulating the selection of discrete values, while the *Video Player* was used for simulating seeking through continuous ranges. Several metrics were gathered to assess the performance including task time, time to target, recognition accuracy, error distance, overshoots and undershoots. To evaluate user experience, participants also completed various questionnaires, and responded to open-ended questions.

The results of the study demonstrated that participants were able to complete tasks more quickly and accurately with *Subtle Swipes* compared to *Large Swipes*. The results from the questionnaires also demonstrated a clear preference for *Subtle Swipes*, with participants consistently reporting that these gestures allowed them to achieve their goals more quickly. They also found it easier to reach precise targets and were more physically comfortable using *Subtle Swipes* for longer periods. Additionally, participants also expressed confidence with using these gestures in public settings.

User Experience Questionnaires (UEQ) were used to evaluate the overall usability and appeal of the two applications. Both applications received positive scores, and users found interactions using *Subtle Swipes* to be easy to learn, efficient, and engaging.

5.6.1 Contributions

This chapter makes the following contributions:

- Provides a design framework for incorporating mmWave radar gesture recognition models into real-time slider-based applications, demonstrated through the development of two applications: a *Photo Scroller* and a *Video Player*.
- Conducts a user study comparing the effectiveness of *Subtle Swipe* gestures in slider-based interactions to traditional *Large Swipes*.
- Demonstrates that *Subtle Swipes* outperform *Large Swipes* in multiple performance

metrics (task time, time to target, error distance, overshoots, and undershoots) and are consistently preferred by users (higher comfort, perceived speed, precision, and social acceptability).

6 | Conclusion

6.1 Introduction

This thesis made the following statement in its Introduction:

> Accurate sensing of subtle mid-air micro-gestures using mmWave radar, and gesture recognition through signal processing and deep learning, enables quick, precise and user-friendly control of virtual interfaces, as demonstrated through empirical trials involving real-time user studies and statistical analysis of metrics such as recognition accuracy, task time, error distance and user experience scores.

The following sections revisit the research questions posed to explore and validate this thesis statement, providing a summary of how each question was addressed and the key findings. The chapter also highlights the main contributions of this research, outlines its limitations, and suggests directions for future work.

6.2 Research Questions

This research explored the following research questions:

RQ1: What types of subtle gestures might be suitable as activation gestures?

RQ2: How can subtle activation gestures be accurately recognized without extensive data acquisition?

RQ3: How accurately can neural networks recognize subtle gestures from mmWave radar data?

RQ4: Do subtle gestures improve task performance and user experience in radar-based interactions involving slider control?

Each of these research questions is addressed in specific chapters of the thesis:

- Chapter 3 addressed **RQ1** and **RQ2**, focusing on the identification of suitable activation gestures and the development of a signal processing pipeline for their recognition.
- Chapter 4 addressed **RQ3** by introducing a dataset for subtle gesture recognition and evaluating deep learning models for gesture classification.
- Chapter 5 addressed **RQ4** by integrating subtle gestures into interactive systems and conducting user studies to assess their effectiveness.

6.2.1 Research Question 1

RQ1: What types of subtle gestures might be suitable as activation gestures?

This question is addressed at the beginning of Chapter 3, which started by analyzing the frequency components of various hand motions. By using frequency analysis techniques such as power spectral density (PSD) and spectrograms, the chapter systematically examined how different hand movements produce distinct frequency signatures. This analysis indicated that while day-to-day hand motions encompass a variety of movements, they typically do not produce significant high-frequency components. Hand movements during routine activities such as cooking, walking, or gesturing while talking predominantly generate low-frequency signals with peaks in the 1-4 Hz range. In contrast, hand motions that produce substantial power in the higher frequency bands (4-12 Hz) require intentional, high-speed, and rhythmic movements. These high-frequency motions are associated with

deliberate actions involving rapid and repetitive movements of small muscle groups, like those in the fingers. Based on these findings, gestures that produced strong high-frequency components were identified as suitable activation gestures because their deliberate and rhythmic nature makes them distinct from everyday motions and less likely to occur accidentally. Accordingly, four candidate gestures were selected—*Finger Taps*, *Finger Rubs*, *Thumb Presses*, and *Pinch Presses*—due to their ability to generate significant high-frequency signals through intentional, repetitive movements.

6.2.2 Research Question 2

> **RQ2:** How can subtle activation gestures be accurately recognized without extensive data acquisition?

This question was addressed in the second half of Chapter 3, which developed and implemented a signal processing pipeline to recognize subtle activation gestures without extensive data acquisition or reliance on machine learning techniques. The pipeline leveraged frequency analysis to distinguish intentional gestures from incidental hand movements by detecting strong high-frequency components (4–12 Hz) in the sensor data, which were unlikely to be produced by casual motions. To evaluate its effectiveness, a user study was conducted using three different sensors—a Google Soli mmWave radar, an Intel D435 camera with MediaPipe hand tracking, and a wrist-mounted SHAKE accelerometer—focusing on metrics such as false activations, time to activation, and user preferences. The results varied across sensors: the Soli sensor effectively filtered out broad hand gestures and had fewer false activations but still encountered some false activations during activities like typing and phone usage; the SHAKE accelerometer achieved the fastest activation times but registered the highest number of false activations due to its high sensitivity and wrist placement; and the Intel D435 had moderate performance with some challenges in hand tracking accuracy. Overall, the pipeline demonstrated that subtle activation gestures could be recognized without extensive data acquisition, but its effectiveness depended on the sensor used and adjustments, such as a multi-trigger

validation mechanism, to minimize false activations.

6.2.3 Research Question 3

RQ3: How accurately can deep learning models recognize subtle gestures from mmWave radar data?

This question was addressed in Chapter 4. A new dataset was developed, comprising 16,000 positive samples of ten distinct subtle gestures and 1,600 negative samples of non-gesture movements, collected from eight participants using a Google Soli radar. Each sample consisted of a sequence of range-Doppler maps (RDMs), capturing the temporal dynamics and motion patterns of the gestures over time. Three neural network architectures—LSTM, CNN-LSTM, and Time-Distributed CNN-LSTM (TD-CNN-LSTM)—were trained and evaluated using Leave-One-Subject-Out Cross-Validation. The CNN-LSTM model achieved the best performance, with an average accuracy of 90.0%, followed by the TD-CNN-LSTM with 85.1%, and the LSTM with 78.5%. These results demonstrated that deep learning models, particularly hybrid architectures combining convolutional and recurrent layers, can accurately recognize subtle gestures from mmWave radar data. The results also indicated that the dataset captured the necessary variations and distinctive features in the RDM sequences for each gesture, enabling the models to learn effective representations and generalize across different users.

6.2.4 Research Question 4

RQ4: Do subtle gestures improve task performance and user experience in radar-based interactive systems?

This question was addressed in Chapter 5, which explored the effectiveness of subtle gestures in radar-based slider control applications. The CNN-LSTM model trained in the previous chapter was integrated into two real-time interactive systems—a *Photo Scroller* for iscrete selection and a *Video Player* for continuous seeking—both controlled using

gestures detected by a Google Soli radar sensor. Three directional swipe gestures were used: traditional *Large Swipes* and two *Subtle Swipes* (*Thumb Swipes* and *Pinch Swipes*). A user study with eight participants compared the performance of these gestures in terms of the models recognition accuracy, task completion time, precision, and user preference. The results showed that the CNN-LSTM model recognized *Subtle Swipes* with similar accuracy to *Large Swipes*, which is notable given the lower intensity and less pronounced radar signatures of *Subtle Swipes*. Participants completed tasks more quickly and accurately using *Subtle Swipes* and expressed a strong preference for them over *Large Swipes*, citing increased comfort, efficiency, and social acceptability. These findings demonstrated that subtle gestures improve task performance and user experience in radar-based interactions involving slider control.

6.3 Contributions

This thesis made contributions to the design, implementation, and evaluation of subtle gesture recognition systems using mmWave radar. Each contribution addressed specific challenges in gesture recognition and interaction design. Design recommendations and a summary of limitations and future work are provided below for each contribution.

6.3.1 Subtle Activation Gesture Recognition Pipeline

This contribution focused on the problem of addressing gesture systems, a fundamental part of interaction; the key issue tackled was the Midas touch problem. First, a set of subtle activation gestures were selected based on spectral analysis of hand motions. By examining the frequency components of various hand movements, four candidate gestures were identified: *Finger Rubs*, *Finger Taps*, *Thumb Presses*, and *Pinch Presses* that produce distinct high-frequency signals. Building on this, a novel *activation gesture recognition pipeline* was developed that used frequency analysis to recognize high-frequency deliberate

gestures from incidental movements. The pipeline aggregates power in specific frequency bands to detect high-frequency gestures while ignoring low-frequency casual motions. The system was evaluated through a user study using three different sensors: a Google Soli mmWave radar, an Intel D435 camera with MediaPipe hand tracking, and a wrist-mounted accelerometer (SHAKE sensor). The results demonstrated that the pipeline effectively ignored broad hand gestures and accurately recognized the selected subtle activation gestures. However, activities involving rapid hand movements like typing, writing, and phone usage occasionally produced false activations due to high-frequency components similar to the activation gestures. To address this, a multi-trigger validation mechanism was introduced, significantly reducing false activations across all sensors.

Design Recommendations

- Use high-frequency, subtle finger gestures for activation due to their low effort, discreetness, and rhythmic nature. The fast rhythmic movements of the fingers generate distinct high-frequency signals that are unlikely to occur frequently during everyday activities, effectively reducing false activations. The rhythmic nature allows users to continue performing the gesture until it is successfully detected.
- Incorporate a multi-trigger validation mechanism into the activation gesture recognition pipeline to reduce false activations during activities involving rapid hand movements. By requiring multiple activation triggers within a brief time window (e.g., two or three triggers within a 2-second window), the likelihood of false activations reduces. However, this introduces a trade-off: increasing the number of required triggers decreases unintended activations but also increases the time to activation, as users need to perform the gesture for longer before it is recognized. Designers should balance the need for minimizing false activations with maintaining a responsive user experience, considering the specific context and user expectations of the application.
- For vision-based hand tracking (e.g. MediaPipe), implement methods to handle occlusions and improve tracking reliability. Incorporate confidence assessments or probabilistic models to reduce false activations caused by tracking errors, especially

in environments where parts of the hand may be obscured.

Limitations and Future Work

- The pipeline produced false activations during activities involving rapid, repetitive hand movements, such as typing, writing, and phone usage. To mitigate this issue, a post-hoc multi-trigger validation mechanism was implemented, requiring multiple activation triggers within a short time window (e.g. two or three triggers within a 2-second window) for confirmation. While this approach eliminated all false activations in most cases in the post-hoc analysis, it was not evaluated in real-time during the user study.
- Sensor-specific challenges were encountered, particularly with the camera-based hand tracking using MediaPipe. The system occasionally struggled with occlusions when participants' hands were partially obscured by objects like pens or phones, which led to erratic jumps in the detected hand landmarks, introducing high-frequency noise that caused false activations. With camera-based tracking, one potential solution is to incorporate uncertainty measures or confidence scores into the hand tracking process. Developing a probabilistic model that assesses the likelihood of valid hand poses could help filter out unreliable data. By setting appropriate confidence thresholds, the system could ignore frames with low-confidence hand detections, reducing the introduction of high-frequency noise due to tracking errors.
- The pipeline was tested in a controlled laboratory environment. While this setting allowed for systematic evaluation, it does not fully represent the variability and unpredictability of real-world conditions. Future research could involve deploying the pipeline 'in the wild' to assess its performance during everyday activities in diverse environments. Evaluating the system in real-world contexts could help identify additional challenges, such as environmental noise, and diverse user behaviors, which could affect the pipelines performance.
- The user study involved eight healthy participants. While this sample size was sufficient to identify significant differences and validate the effectiveness of the pipeline,

a larger participant group would provide greater statistical power. However, since all participants were healthy adults, increasing the number of similar participants might not reveal substantially new insights. Instead, future studies should focus on including more diverse groups, such as senior citizens, who may have difficulty performing subtle gestures like *Tremor-Inducing Pressure Gestures*. Evaluating the pipeline's performance across such populations could highlight specific limitations or necessary design adaptations to improve accessibility and inclusivity.

6.3.2 Subtle Gesture Recognition Using Neural Networks

This contribution focused on the challenge of recognizing subtle mid-air gestures from mmWave radar data—a relatively unexplored area compared to macro-gesture recognition. To address this, a new dataset was created, comprising radar data of 10 subtle gestures captured using a Google Soli sensor. The gestures were selected based on a "virtual tools" gesture language, which mimics real-world interactions such as swiping, pinching, and rotating, and also used proprioception and natural haptic feedback. Three neural network architectures—LSTM, CNN-LSTM, and TD-CNN-LSTM—were trained and evaluated using Leave-One-Subject-Out Cross-Validation (LOSO-CV). The CNN-LSTM model achieved the highest accuracy of 90.0%, demonstrating that hybrid deep learning models can accurately recognize subtle gestures from mmWave radar data. This also validated the dataset which captured the necessary variations and distinctive features of each gesture, enabling the models to learn meaningful representations and generalize well across different users.

Design Recommendations

- Choose hybrid architectures like CNN-LSTM for gesture recognition to effectively capture both spatial and temporal features. Additionally, consider factors such as model size and inference time to ensure suitability for the deployment environment.

Limitations and Future Work

- The subtle gestures dataset was created using the short-range settings of the Soli radar, limiting the maximum sensing range to 20cm. Future work could explore subtle gesture detection from longer ranges by utilizing more sophisticated radar systems, such as those that support Multiple Input Multiple Output (MIMO) setups, which have fine spatial resolution and can detect small movements over larger distances.
- If detecting subtle gestures from longer ranges is feasible, future work could focus on creating a new dataset of long-range RDM sequences. Transfer learning techniques could be employed to adapt existing gesture detection and recognition models from this research to the new data, reducing the need for extensive new data collection.
- The dataset was created with data from eight healthy, right-handed participants. While LOSO-CV demonstrated that the trained models could generalize across participants, future expansions with a larger or more diverse participant group (e.g., older adults, individuals with hand tremors, or left-handed participants) could capture a broader range of real-world gesture variations. This would enable the models to better adapt to diverse user needs and reduce potential biases in the training data.
- Another avenue for future work could leverage recent advancements in generative models for radar data [74]. The work by Tonolini et al. demonstrated the ability to generate synthetic RDM data given a hand pose captured through an OptiTrack motion tracking system. Future work could try to use this approach to generate synthetic RDM data for subtle gestures. This approach could allow researchers to expand or even create new datasets without needing additional participants or prolonged recording sessions.

6.3.3 Evaluating Subtle Gestures in Real-Time Applications

While many mmWave radar gesture recognition models have been developed and evaluated offline, there is a notable lack of research testing the application of gestures—particularly subtle gestures—in real-time applications. This contribution addressed this gap by assessing

the effectiveness and user experience of subtle gestures in real-time applications involving slider-based controls. Sliders were chosen because they are fundamental interactive components widely used for tasks such as adjusting settings, navigating through content, and making selections. By integrating the previously developed CNN-LSTM gesture recognition model into two slider-based interactive applications, the research investigated whether subtle gestures could improve task performance compared to traditional large gestures. A user study with eight participants demonstrated that subtle gestures allowed users to complete tasks more quickly and with greater precision. Participants also expressed a strong preference for subtle gestures, citing increased comfort, efficiency, and perceived social acceptability.

Design Recommendations

- Use gestures like *Thumb Swipes* and *Pinch Swipes* (Figure 4.1) for slider control. These gestures align well with the interaction metaphor of virtual tools for slider control, where the index finger can be imagined as the slider itself, and the thumb moving along the index finger mimics the action of adjusting the slider handle. Such interaction metaphors can be generalised to other interface components—for example, *Index Finger Rotations* can represent turning a virtual dial, while *Open* and *Close Pinch* gestures can support zooming in and out, respectively.
- Implement speed control using hand distance from the radar, allowing users to dynamically adjust the speed of continuous interactions. This enhancement enables quicker and more precise adjustments depending on proximity. While demonstrated in this study for slider speed control, the same principle can generalize to other interactions—for instance, controlling dial rotation speed after a *Finger Rotation*, or adjusting zoom rate after *Open* and *Close Pinch* gestures. By remaining in a closed interaction loop, users gain fine control over the intensity or velocity of the action being performed.
- Combine spatial activation zones with rhythmic high-frequency gestures (e.g., *Thumb Presses* or *Finger Rubs*) to "grab" the slider handle in real applications. In the user

study, the slider handle was acquired simply by bringing the hand into the radar's sensing range—an approach suitable for controlled experimental settings. However, for real-world use, requiring deliberate activation gestures to be performed within the activation zone would offer both spatial and intentional context. This approach can also generalize to other interaction metaphors—for example, requiring an activation gesture to engage with a virtual dial or initiate a zoom action ensures that subsequent control inputs are intentional.

Limitations and Future Work

- In the slider control user study, participants reported that they would feel comfort using subtle gestures in public, however, this was not validated. Future work could address this by conducting 'in the wild' evaluations with Soli-embedded devices like smartwatches to assess recognition performance under real-world conditions, considering factors like environmental noise, diverse user behaviors, and device placement.
- Subtle gestures, while offering privacy and discretion to the user, may appear deceptive to observers, raising ethical concerns in social interactions. For instance, a TV channel might be changed surreptitiously, leaving family members puzzled, or a person might engage with their device during a dinner date without their companion's knowledge. Future research could explore design interventions to maintain the benefits of subtlety while ensuring relevant observers remain aware that an interaction is taking place. Appropriate feedback mechanisms—visual, auditory, or other unobtrusive cues—could help balance the need for discretion with the desire for transparency and ethical, socially acceptable behavior.
- The user study focused on a limited set of gestures, suitable for slider control but not representative of the full range of potential interactions. Future work could integrate the broader set of gestures—such as *Index Finger Rotations* for dial manipulation or *Open* and *Close Pinch* gestures for zooming—into other real-time applications. By evaluating these new gestures in context, and comparing them directly to familiar

touchscreen gestures, future work can continue to explore "virtual tools" interactions using mmWave radar.

- This study's sample comprised eight healthy participants, and some had prior experience from earlier experiments with radar gestures. While the results demonstrated consistent trends, the small sample size and participant overlap reduce statistical power. Future work could recruit larger, more diverse groups of participants, including those who are older, and have no prior experience with radar-based gestures. This would help capture a broader range of user behaviors and increase the ecological validity of the results.
- This study focused on a single-device context with short-range sensing, which limits gesture input to the immediate vicinity of the device. For example, bringing a hand close to a thermostat or smartwatch and performing a gesture within its sensing range ensures that only the intended device responds. However, if the gesture recognition systems developed in this thesis are extended to support long-range sensing, multiple radar-embedded devices in the same space (e.g., a TV and a smart speaker) may simultaneously detect the same gesture, leading to unintended activations. Future work should explore spatial disambiguation strategies—such as device-specific activation zones, beamforming, or gesture differentiation based on rhythm or frequency characteristics—to ensure reliable, context-aware targeting in shared environments.

6.4 Summary

This thesis addressed usability issues with macro-gestures by focusing on interaction using subtle mid-air gestures. It developed and evaluated techniques that enable users to interact with mmWave radar embedded systems using discrete and low-effort hand gestures. The research introduced methods for addressing gesture systems, recognizing a diverse set of subtle gestures from mmWave radar data, and evaluated these techniques in real-time

through user studies. The results demonstrate that the developed techniques are successful, highly practical, and have the potential to be implemented across a wide range of gesture systems.

A scenario was presented in the Introduction of this thesis to illustrate the usability challenges associated with macro-gestures. These included physical discomfort, unintended activations stemming from the Midas Touch problem, and social acceptability concerns. The revised scenario below demonstrates how the systems developed in this thesis address these challenges and lead to better outcomes.

> Preethi wants to watch a program on Netflix and needs to browse through the catalog. She addresses the TV by bringing her hand close to the Soli sensor and performs a *Finger Rub* activation gesture. Since the gesture is subtle and generates a high-frequency signal, it is only recognized by the TV's activation system while remaining undetected by other smart devices in the room. With the system now activated, Preethi browses through the Netflix catalog using *Thumb* and *Pinch Swipes*. The low-effort nature of these gestures allows her to navigate comfortably without tiring her hand or arm. She also feels comfortable using these gestures when her friends are over and no longer feels self-conscious about looking awkward.

A | Appendix A

A.1 Spectral Profiles of Casual Hand Motions

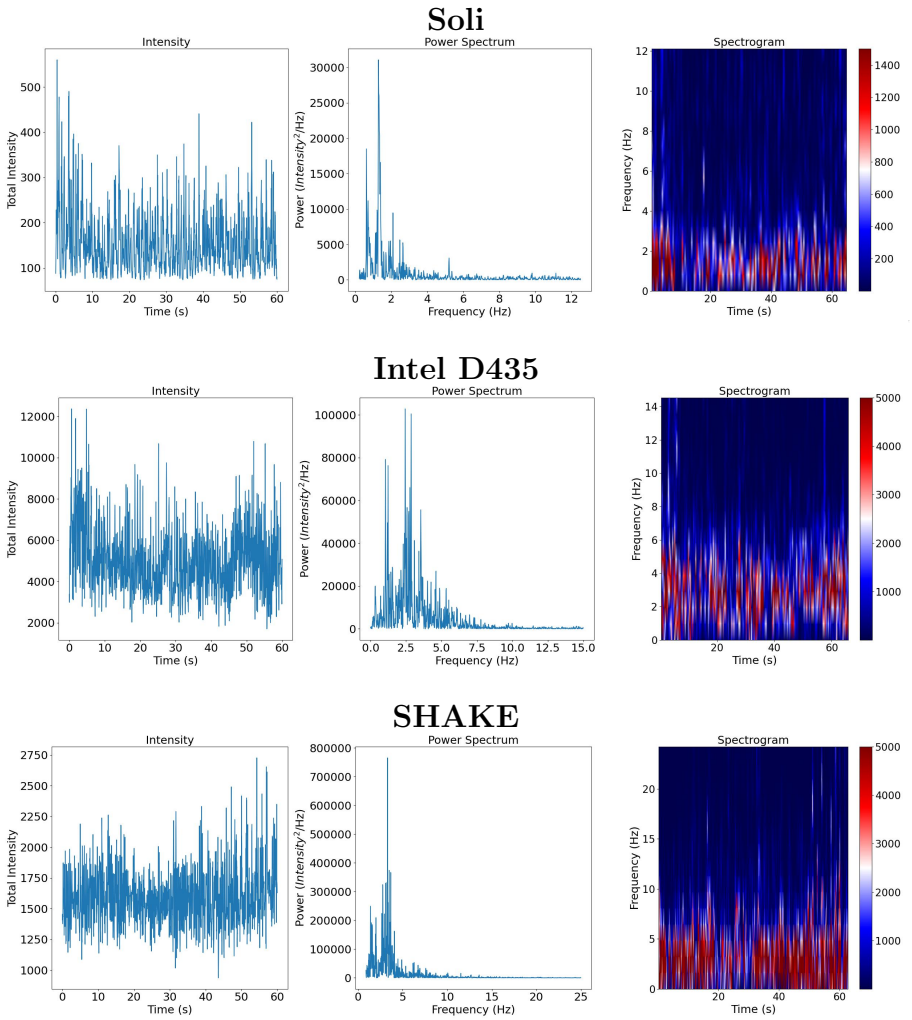


Figure A.1: Spectral profiles of *broad hand motions* across the three sensors (top: Soli, middle: Intel D435, bottom: SHAKE), generated from composite signals created by aligning and averaging signal intensities across all participants. The PSD for these gestures consistently exhibit strong peaks in the low-frequency band (0–4Hz). In the spectrogram, bright bands of activity are consistent below 4Hz for all three sensors.

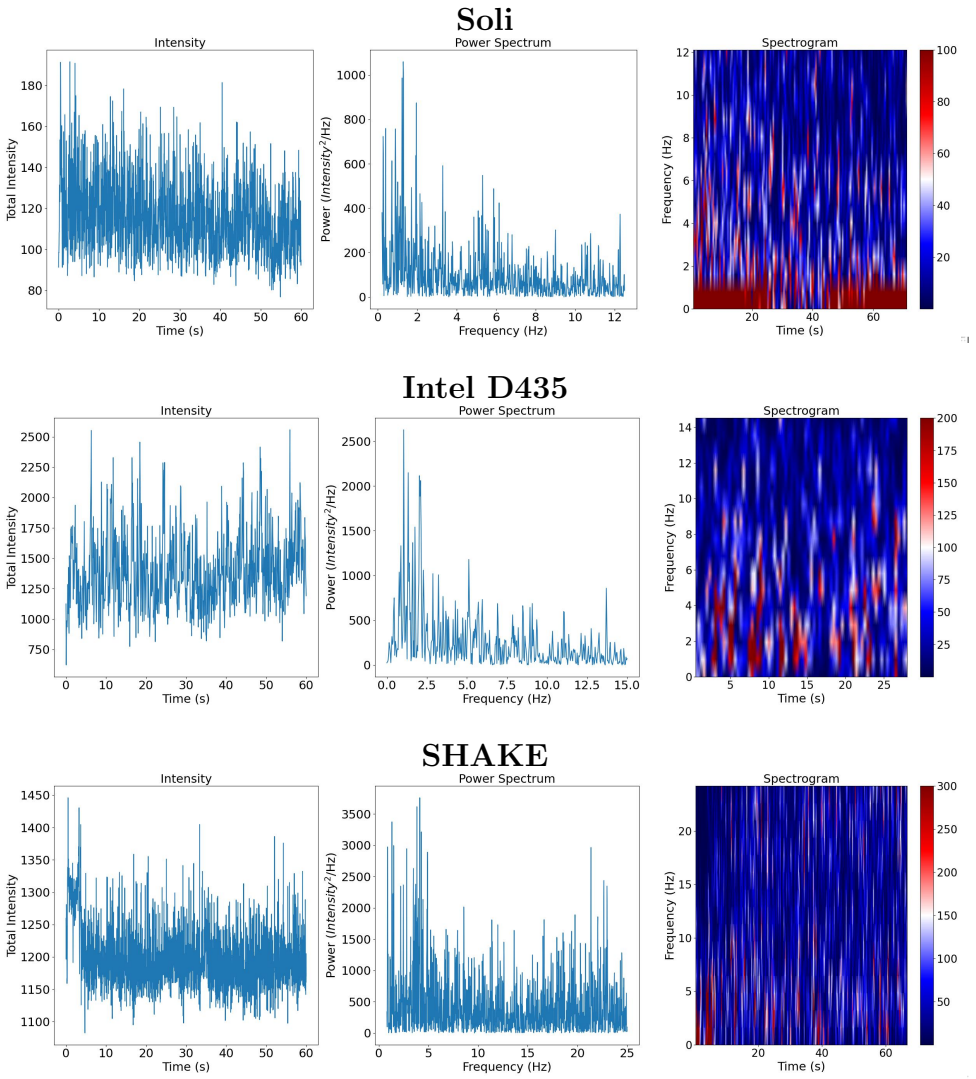


Figure A.2: Spectral profiles of *typing task* across the three sensors (top: Soli, middle: Intel D435, bottom: SHAKE), generated from composite signals created by aligning and averaging signal intensities across all participants. The PSDs for all sensors reveal multiple peaks at higher frequencies (between 4–12Hz), reflecting the rapid finger movements characteristic of typing. The spectrograms also shows bursts of strong energy concentrated in these higher-frequency regions.

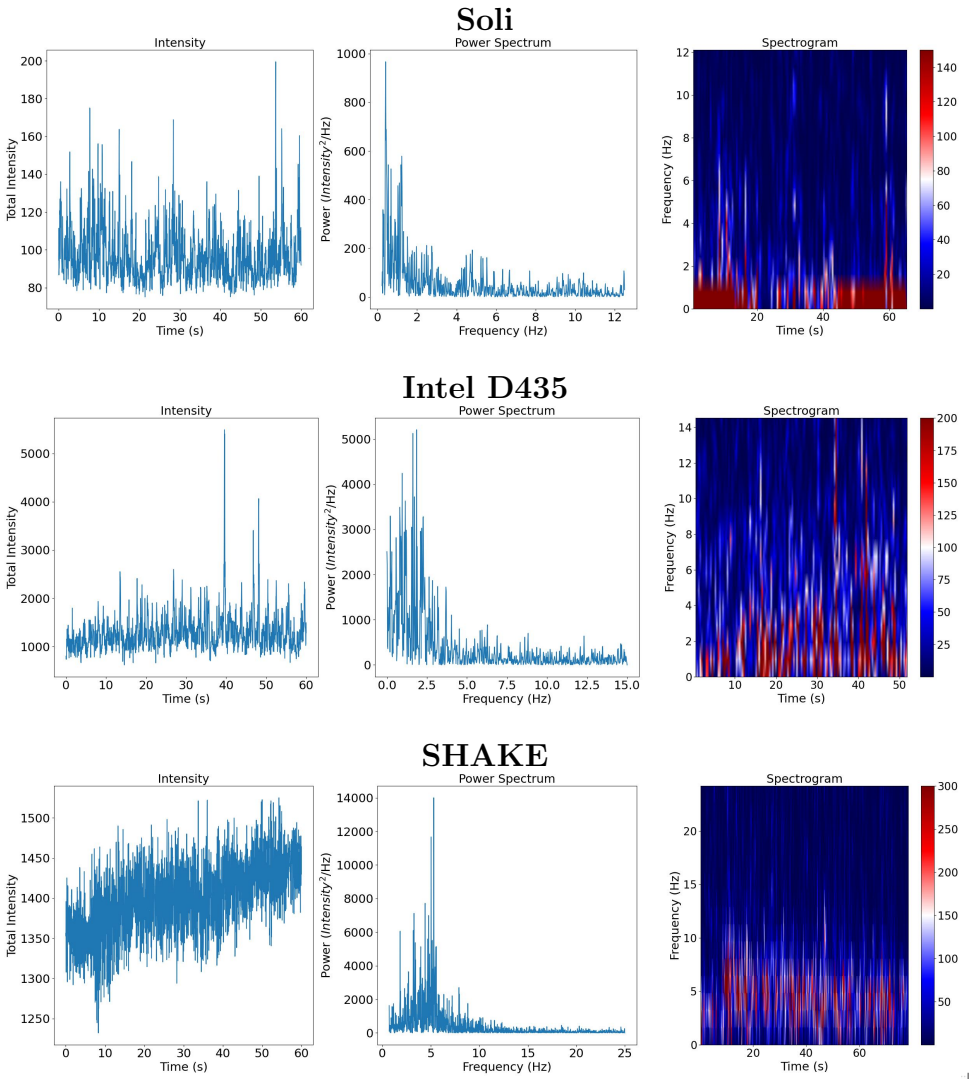


Figure A.3: Spectral profiles of *writing task* across the three sensors (top: Soli, middle: Intel D435, bottom: SHAKE), generated from composite signals created by aligning and averaging signal intensities across all participants. Most of the activity is concentrated below 4Hz, as seen by the peaks in the PSDs in lower frequencies and bright, continuous bands in the spectrograms. However, occasional bursts of high-frequency activity are present, particularly in the spectrogram for the Intel D435.

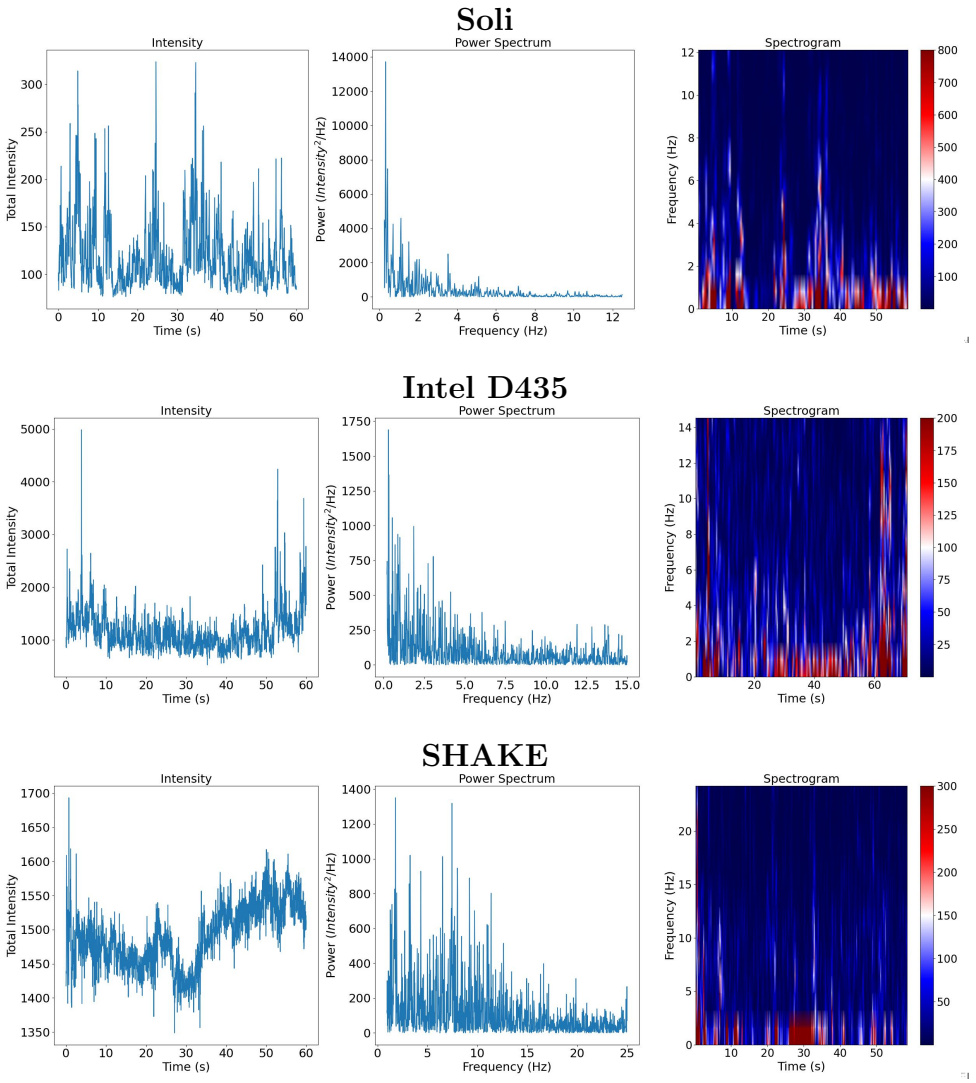


Figure A.4: Spectral profiles of *phone usage task* across the three sensors (top: Soli, middle: Intel D435, bottom: SHAKE), generated from composite signals created by aligning and averaging signal intensities across all participants. The PSDs for Soli and Intel D435 predominantly show strong activity below 4Hz. However, in the Intel D435 spectrogram, brief bursts of high-frequency energy appear, while the SHAKE data exhibit more pronounced peaks in the 4–12Hz range.

A.2 Pseudocode for the Activation Gesture Detection Pipeline with Multi-Trigger Validation

Data: Sensor data stream, sliding window size (*history_length* for 1 second), high-pass filter parameters, frequency bands (0-4 Hz, 4-12 Hz), validation window (2 seconds)
Result: Detected activation gestures
begin
  Initialize *total_intensities* deque with max length *history_length*
  Initialize $filtered\_intensities$ deque with max length *history_length*
  Initialize *triggers* deque for recent timestamps within 2 seconds
  while *running* **do**
    Retrieve raw data from sensor stream
    Process raw data to calculate *signal_intensity* representing motion intensity
    Append *signal_intensity* to *total_intensities*
    if *total_intensities has sufficient data for 1 second* **then**
      $filtered\_data \longleftarrow$ Apply high-pass filter to *total_intensities*
      $(low\_frequency\_power, high\_frequency\_power) \longleftarrow$ Calculate PSD and aggregate power in frequency bands
      if $high\_frequency\_power > low\_frequency\_power$ **then**
        Record current timestamp in *triggers*
        while *timestamps in triggers are older than 2 seconds* **do**
          Remove the oldest timestamp from *triggers*
        end
        if *triggers contains 3 timestamps* **then**
          Confirm activation gesture and clear *triggers*
          Clear *total_intensities* and $filtered\_intensities$
        end
      end
    end
  end
end

B | Appendix B

B.1 Deriving Hand Distance from Radar Data

The derivation of hand distance from the radar data is a critical component of the continuous seeking task, enabling control of the slider based on the user's hand position relative to the Google Soli sensor. This process involves several steps to accurately capture and interpret the radar signals.

Blob Detection The first step involves identifying significant reflections that correspond to the hand. This is achieved through the use of a blob detection algorithm known as the Laplacian of Gaussian (LoG) [1]. The LoG method is particularly effective for identifying blobs, which are contiguous regions in the range Doppler map that stand out due to their higher signal intensity, indicating a potential target object like a hand.

Subpixel Position Refinement Once a potential hand position is detected, the exact center of this detected blob is refined to subpixel accuracy. It involves fitting a quadratic polynomial to the intensity values around the detected center, adjusting the estimated position based on the polynomial's maximum point, which provides the most accurate representation of the hand's central position.

Distance Calculation The refined position coordinates are then used to calculate the physical distance from the radar sensor. The dimensions of the range Doppler map, which are 8 rows and 64 columns, allow for this conversion from pixel coordinates to actual distance measurements. Each pixel in the map corresponds to a specific portion of the

[1] "Blob Detection Using Laplacian of Gaussian." Available at `https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_blob.html`

radar's field of view, with coordinates scaled to reflect this mapping accurately. The distance is calculated based on the displacement of the hand's position from the range Doppler map's pixel grid, using the normalized positions $x = \frac{c}{64}$ and $y = \frac{r}{8}$, where c and r are the column and row indices of the detected blob's center. This method allows for precise tracking of hand movements, mapping them accurately within the spatial dimensions that the radar covers.

B.2 User Experience Questionnaire

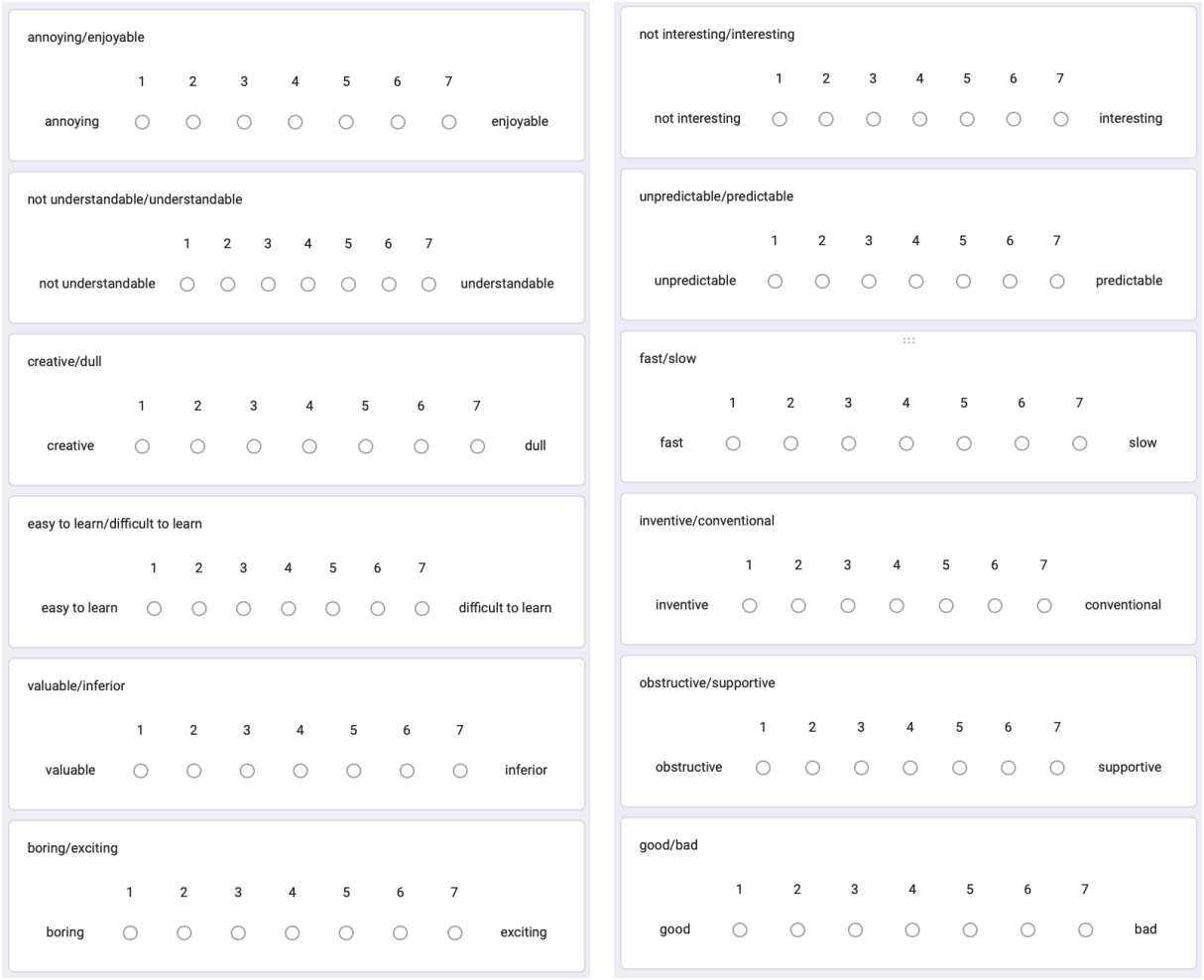


Figure B.1: User Experience Questionnaire (1/3)

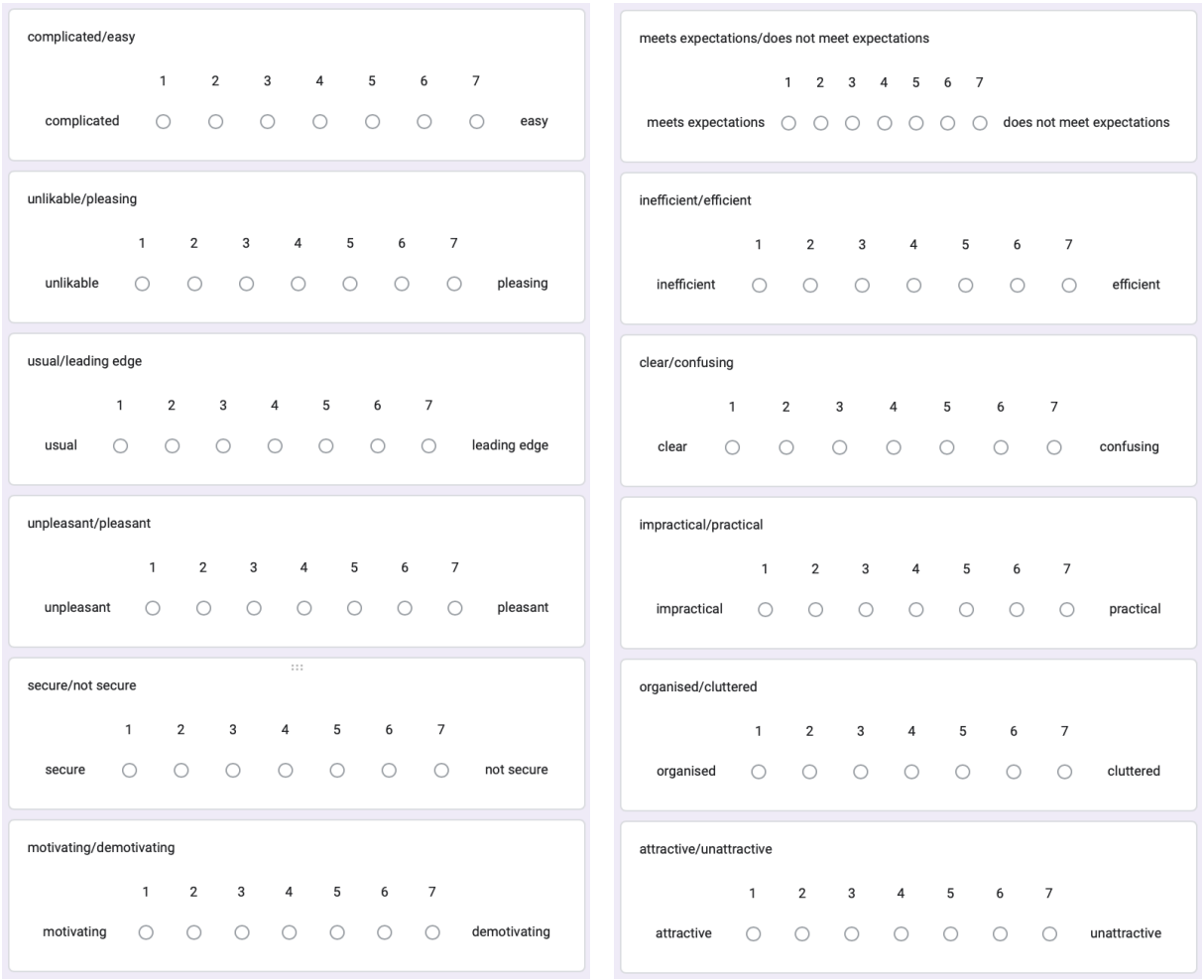


Figure B.2: User Experience Questionnaire (2/3)

Figure B.3: User Experience Questionnaire (3/3)

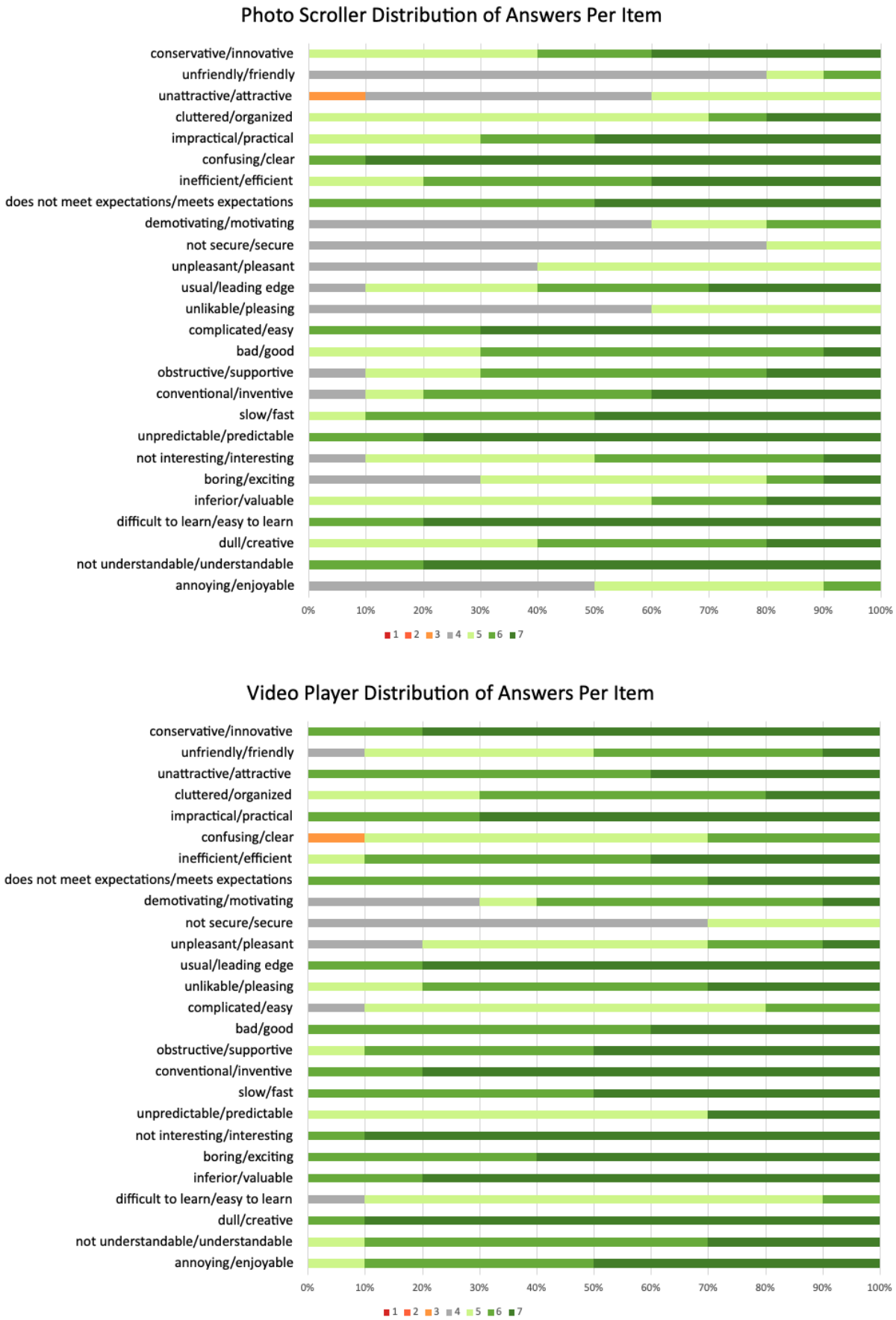


Figure B.4: Distribution of UEQ Responses for Photo Scroller and Video Player.

Bibliography

[1] Abdelnasser, H., Youssef, M. and Harras, K. A. [2015], 'WiGest: A Ubiquitous WiFi-Based Gesture Recognition System', *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)* pp. 1472–1480.

[2] Ahmed, S., Kallu, K. D., Ahmed, S. and Cho, S. H. [2021], 'Hand Gestures Recognition Using Radar Sensors for Human-Computer-Interaction: A Review', *Remote Sensing* **13**(3), 527.

[3] Ahmed, S., Wang, D., Park, J. and Cho, S. H. [2021], 'UWB-Gestures, A Public Dataset of Dynamic Hand Gestures Acquired Using Impulse Radar Sensors', *Scientific Data* **8**(102), 1–9.

[4] Al-Eidan, R., Al-Khalifa, H. and Al-Salman, A. [2018], 'A Review of Wrist-Worn Wearable: Sensors, Models, and Challenges', *Journal of Sensors* pp. 1–20.

[5] Ali, A., Parida, P., Va, V., Ni, S., Nguyen, K. N., Ng, B. L. and Zhang, J. C. [2022], 'End-to-End Dynamic Gesture Recognition Using MmWave Radar', *IEEE Access* **10**, 88692–88706.

[6] Ashbrook, D., Baudisch, P. and White, S. [2011], 'Nenya: Subtle and Eyes-Free Mobile Input With a Magnetically-Tracked Finger Ring', *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* p. 2043–2046.

[7] Baudel, T. and Beaudouin-Lafon, M. [1993], 'Charade: Remote Control of Objects Using Free-Hand Gestures', *Communications of the ACM* **36**(7), 28–35.

[8] Bellotti, V., Back, M., Edwards, W. K., Grinter, R. E., Henderson, A. and Lopes, C. [2002], 'Making Sense of Sensing Systems: Five Questions for Designers and Researchers', *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* p. 415–422.

[9] Bolt, R. A. [1980], '"Put-That-There": Voice and Gesture at the Graphics Interface', *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques* **14**(3), 262–270.

[10] Chan, E., Seyed, T., Stuerzlinger, W., Yang, X.-D. and Maurer, F. [2016], 'User Elicitation on Single-hand Microgestures', *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* p. 3403–3414.

[11] Chan, L., Liang, R.-H., Tsai, M.-C., Cheng, K.-Y., Su, C.-H., Chen, M. Y., Cheng, W.-H. and Chen, B.-Y. [2013], 'FingerPad: Private and Subtle Interaction Using Fingertips', *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* p. 255–260.

[12] Chen, K.-Y., Ashbrook, D., Goel, M., Lee, S.-H. and Patel, S. [2014], 'AirLink: Sharing Files Between Multiple Devices Using In-Air Gestures', *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* p. 565–569.

[13] Choi, J.-W., Park, C.-W. and Kim, J.-H. [2022], 'FMCW Radar-Based Real-Time Hand Gesture Recognition System Capable of Out-of-Distribution Detection', *IEEE Access* **10**, 87425–87434.

[14] Choi, J.-W., Ryu, S.-J. and Kim, J.-H. [2019], 'Short-Range Radar Based Real-Time Hand Gesture Recognition Using LSTM Encoder', *IEEE Access* **7**, 33610–33618.

[15] Cohen, L. [1994], *Time Frequency Analysis: Theory and Applications*, Prentice Hall.

[16] Costanza, E., Inverso, S. A., Pavlov, E., Allen, R. and Maes, P. [2006], 'Eye-Q: Eyeglass Peripheral Display for Subtle Intimate Notifications', *Proceedings of the 8th Conference on Human-Computer Interaction with Mobile Devices and Services* p. 211–218.

[17] Costanza, E., Inverso, S. and Allen, R. [2005], 'Toward Subtle Intimate Interfaces for Mobile Devices Using an EMG Controller', *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* p. 481–489.

[18] Dementyev, A. and Paradiso, J. A. [2014], 'WristFlex: Low-Power Gesture Input With Wrist-Worn Pressure Sensors', *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* p. 161–166.

[19] Dong, X., Zhao, Z., Wang, Y., Zeng, T., Wang, J. and Sui, Y. [2021], 'FMCW Radar-Based Hand Gesture Recognition Using Spatiotemporal Deformable and Context-Aware Convolutional 5-D Feature Representation', *IEEE Transactions on Geoscience and Remote Sensing* **60**, 1–1.

[20] Elble, R. J. [2017], Tremor, *in* 'Neuro-Geriatrics: A Clinical Manual', Springer International Publishing, pp. 311–326.

[21] Fayyad, J., Jaradat, M. A. K., Gruyer, D. and Najjaran, H. [2020], 'Deep Learning Sensor Fusion for Autonomous Vehicle Perception and Localization: A Review', *Sensors* **20**(15), 20.

[22] Freeman, E., Brewster, S. and Lantz, V. [2016], 'Do That, There: An Interaction Technique for Addressing In-Air Gesture Systems', *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* p. 2319–2331.

[23] Freeman, E., Griffiths, G. and Brewster, S. A. [2017], 'Rhythmic Micro-Gestures: Discreet Interaction On-the-Go', *Proceedings of the 19th ACM International Conference on Multimodal Interaction* p. 115–119.

[24] Golod, I., Heidrich, F., Möllering, C. and Ziefle, M. [2013], 'Design Principles of Hand Gesture Interfaces for Microinteractions', *Proceedings of the 6th International Conference on Designing Pleasurable Products and Interfaces* p. 11–20.

[25] Gong, J., Zhang, Y., Zhou, X. and Yang, X.-D. [2017], 'Pyro: Thumb-Tip Gesture Recognition Using Pyroelectric Infrared Sensing', *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* p. 553–563.

[26] Gupta, S., Morris, D., Patel, S. and Tan, D. [2012], 'SoundWave: Using the Doppler Effect to Sense Gestures', *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* p. 1911–1914.

[27] Hajika, R., Gunasekaran, T. S., Haigh, C. D. S. Y., Pai, Y. S., Hayashi, E., Lien, J., Lottridge, D. and Billinghurst, M. [2024], 'RadarHand: A Wrist-Worn Radar for On-Skin Touch-Based Proprioceptive Gestures', *ACM Transactions on Computer-Human Interaction* **31**(2).

[28] Hansson, R. and Ljungstrand, P. [2000], 'The Reminder Bracelet: Subtle Notification Cues for Mobile Devices', *CHI '00 Extended Abstracts on Human Factors in Computing Systems* p. 323–324.

[29] Hayashi, E., Lien, J., Gillian, N., Giusti, L., Weber, D., Yamanaka, J., Bedal, L. and Poupyrev, I. [2021], 'RadarNet: Efficient Gesture Recognition Technique Utilizing a Miniature Radar Sensor', *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* p. 1–14.

[30] Hazra, S. and Santra, A. [2018], 'Robust Gesture Recognition Using Millimetric-Wave Radar System', *IEEE Sensors Letters* **2**(4), 1–4.

[31] He, W., Wu, K., Zou, Y. and Ming, Z. [2015], 'WiG: WiFi-Based Gesture Recognition System', *Proceedings of the 24th International Conference on Computer Communication and Networks* pp. 1–7.

[32] Hudson, S. E., Harrison, C., Harrison, B. L. and LaMarca, A. [2010], 'Whack Gestures: Inexact and Inattentive Interaction With Mobile Devices', *Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction* p. 109–112.

[33] Jing, L., Cheng, Z., Zhou, Y., Wang, J. and Huang, T. [2013], 'Magic Ring: A Self-Contained Gesture Input Device on Finger', *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia* p. 1–4.

[34] Kerber, F., Schardt, P. and Löchtefeld, M. [2015], 'WristRotate: A Personalized Motion Gesture Delimiter For Wrist-Worn Devices', *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia* p. 218–222.

[35] Kjeldsen, R. and Hartman, J. [2001], 'Design Issues for Vision-Based Computer Interaction Systems', *Proceedings of the 2001 Workshop on Perceptive User Interfaces* p. 1–8.

[36] Kratz, S. and Wiese, J. [2016], 'GestureSeg: Developing a Gesture Segmentation System Using Gesture Execution Phase Labeling by Crowd Workers', *Proceedings of the 8th ACM SIGCHI Symposium on Engineering Interactive Computing Systems* p. 61–72.

[37] Krizhevsky, A., Sutskever, I. and Hinton, G. E. [2017], 'ImageNet Classification With Deep Convolutional Neural Networks', *Communications of the ACM* **60**(6), 84–90.

[38] Lantz, V. and Murray-Smith, R. [2004], Rhythmic interaction with a mobile device, *in* 'Proceedings of the third Nordic conference on Human-computer interaction', pp. 97–100.

[39] Laugwitz, B., Held, T. and Schrepp, M. [2008], Construction and Evaluation of a User Experience Questionnaire, *in* 'HCI and Usability for Education and Work', Springer Berlin Heidelberg, pp. 63–76.

[40] Lee, J., Aggarwal, S., Wu, J., Starner, T. and Woo, W. [2019], 'SelfSync: Exploring Self-Synchronous Body-Based Hotword Gestures for Initiating Interaction', *Proceedings of the 2019 ACM International Symposium on Wearable Computers* p. 123–128.

[41] Li, J. and Stoica, P. [2009], 'MIMO Radar Signal Processing', *Wiley-IEEE Press* .

[42] Lien, J., Gillian, N., Karagozler, M. E., Amihood, P., Schwesig, C., Olson, E., Raja, H. and Poupyrev, I. [2016], 'Soli: Ubiquitous Gesture Sensing with Millimeter Wave Radar', *ACM Transactions on Graphics* **35**(4), 1–19.

[43] Ling, K., Dai, H., Liu, Y., Liu, A. X., Wang, W. and Gu, Q. [2022], 'UltraGesture: Fine-Grained Gesture Sensing and Recognition', *IEEE Transactions on Mobile Computing* pp. 2620–2636.

[44] Liu, H., Zhou, A., Dong, Z., Sun, Y., Zhang, J., Liu, L., Ma, H., Liu, J. and Yang, N. [2022], 'M-Gesture: Person-Independent Real-Time In-Air Gesture Recognition Using Commodity Millimeter Wave Radar', *IEEE Internet of Things Journal* **9**(5), 3397–3415.

[45] Liu, M., Nancel, M. and Vogel, D. [2015], 'Gunslinger: Subtle Arms-down Mid-Air Interaction', *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology* p. 63–71.

[46] Liu, Y., Wang, Y., Liu, H., Zhou, A., Liu, J. and Yang, N. [2020], 'Long-Range Gesture Recognition Using Millimeter Wave Radar', *Green, Pervasive, and Cloud Computing* pp. 30–44.

[47] Liu, Z., Liu, H. and Ma, C. [2022], 'A Robust Hand Gesture Sensing and Recognition Based on Dual-Flow Fusion With FMCW Radar', *IEEE Geoscience and Remote Sensing Letters* **19**, 1–5.

[48] MathWorks [Accessed: 2024-10-14], 'Range-Doppler Response - MathWorks Documentation'. `https://uk.mathworks.com/help/phased/ug/range-doppler-response.html`.

[49] Mistry, P. and Maes, P. [2009], 'SixthSense: A Wearable Gestural Interface', *ACM SIGGRAPH ASIA 2009 Sketches* p. 11.

[50] Mitchell, K., Kassem, K., Kaul, C., Kapitany, V., Binner, P., Ramsay, A., Faccio, D. and Murray-Smith, R. [2023], 'mmSense: Detecting Concealed Weapons with a Miniature Radar Sensor', *IEEE International Conference on Acoustics, Speech and Signal Processing* pp. 1–5.

[51] Molchanov, P., Gupta, S., Kim, K. and Pulli, K. [2015], 'Short-Range FMCW Monopulse Radar for Hand-Gesture Sensing', *2015 IEEE Radar Conference* pp. 1491–1496.

[52] Nandakumar, R., Iyer, V., Tan, D. and Gollakota, S. [2016], 'FingerIO: Using Active Sonar for Fine-Grained Finger Tracking', *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* p. 1515–1525.

[53] Nguyen, V., Rupavatharam, S., Liu, L., Howard, R. and Gruteser, M. [2019], 'HandSense: Capacitive Coupling-Based Dynamic, Micro Finger Gesture Recognition', *Proceedings of the 17th Conference on Embedded Networked Sensor Systems* p. 285–297.

[54] Nogales, R. and Benalcázar, M. [2021], 'Hand Gesture Recognition Using Machine Learning and Infrared Information: A Systematic Literature Review', *International Journal of Machine Learning and Cybernetics* **12**, 2859–2886.

[55] Pan, M., Tang, Y. and Li, H. [2023], 'State-of-the-Art in Data Gloves: A Review of Hardware, Algorithms, and Applications', *IEEE Transactions on Instrumentation and Measurement* **72**, 1–15.

[56] Pohl, H., Muresan, A. and Hornbæk, K. [2019], 'Charting Subtle Interaction in the HCI Literature', *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* p. 1–15.

[57] Pohl, H. and Murray-Smith, R. [2013], 'Focused and Casual Interactions: Allowing Users to Vary Their Level of Engagement', *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* p. 2223–2232.

[58] Poupyrev, I., Gong, N.-W., Fukuhara, S., Karagozler, M. E., Schwesig, C. and Robinson, K. E. [2016], 'Project jacquard: Interactive digital textiles at scale', *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* p. 4216–4227.

[59] Proske, U. and Gandevia, S. C. [2012], 'The Proprioceptive Senses: Their Roles in Signaling Body Shape, Body Position and Movement, and Muscle Force', *Physiological Reviews* **92**(4), 1651–1697.

[60] Pu, Q., Gupta, S., Gollakota, S. and Patel, S. [2013], 'Whole-home gesture recognition using wireless signals', *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking* p. 27–38.

[61] Rao, S., Ahmad, A., Roh, J. C. and Bharadwaj, S. [2017], '77GHz Single Chip Radar Sensor Enables Automotive Body and Chassis Applications', Texas Instruments. Available online: `https://www.ti.com/lit/wp/spry315/spry315.pdf?ts=1722175009171` (accessed on 29 July 2024).

[62] Ravindran, A. M. [2024], 'Soli Subtle Gesture Dataset'. OSF.IO.
URL: *https://doi.org/10.17605/OSF.IO/EVUDB*

[63] Rawat, W. and Wang, Z. [2017], 'Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review', *Neural Computation* **29**, 2352–2449.

[64] Ritchie, M., Capraru, R. and Fioranelli, F. [2020], 'Dop-NET: A Micro-Doppler Radar Data Challenge', *Electronics Letters* **56**(11), 568–570.

[65] Ruan, W., Sheng, Q. Z., Yang, L., Gu, T., Xu, P. and Shangguan, L. [2016], 'AudioGest: Enabling Fine-Grained Hand Gesture Detection by Decoding Echo Signal',

Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing p. 474–485.

[66] Schwarz, J., Marais, C. C., Leyvand, T., Hudson, S. E. and Mankoff, J. [2014], 'Combining Body Pose, Gaze, and Gesture to Determine Intention to Interact in Vision-Based Interfaces', *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* p. 3443–3452.

[67] Shannon, C. [1949], 'Communication in the Presence of Noise', *Proceedings of the Institute of Radio Engineers* **37**(1), 10–21.

[68] Shastri, A., Valecha, N., Bashirov, E., Tataria, H., Lentmaier, M., Tufvesson, F., Rossi, M. and Casari, P. [2022], 'A Review of Millimeter Wave Device-Based Localization and Device-Free Sensing Technologies and Applications', *IEEE Communications Surveys Tutorials* **24**(3), 1708–1749.

[69] Shen, X., Zheng, H., Feng, X. and Hu, J. [2022], 'ML-HGR-Net: A Meta-Learning Network for FMCW Radar Based Hand Gesture Recognition', *IEEE Sensors Journal* **22**(11), 10808–10817.

[70] Singh, A., Rehman, S. U., Yongchareon, S. and Chong, P. H. J. [2021], 'Multi-Resident Non-Contact Vital Sign Monitoring Using Radar: A Review', *IEEE Sensors Journal* **21**(4), 4061–4084.

[71] Sørensen, T., Andersen, O. D. and Merritt, T. [2015], '"Tangible Lights": In-Air Gestural Control of Home Lighting', *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction* p. 727–732.

[72] Sridharan, S., Bailey, R., McNamara, A. and Grimm, C. [2012], 'Subtle Gaze Manipulation for Improved Mammography Training', *Proceedings of the Symposium on Eye Tracking Research and Applications* p. 75–82.

[73] Stimson, G. W. [2014], *Introduction to Airborne Radar - Third Edition*, SciTech.

[74] Tonolini, F., Radford, J., Turpin, A., Faccio, D. and Murray-Smith, R. [2020], 'Variational Inference for Computational Imaging Inverse Problems', *Journal of Machine Learning Research* **21**(179), 1–46.

[75] Tsai, H.-R., Hsiu, M.-C., Hsiao, J.-C., Huang, L.-T., Chen, M. and Hung, Y.-P. [2016], 'TouchRing: Subtle and Always-Available Input Using a Multi-Touch Ring', *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct* p. 891–898.

[76] Tsai, H.-R., Wu, C.-Y., Huang, L.-T. and Hung, Y.-P. [2016], 'ThumbRing: Private Interactions Using One-Handed Thumb Motion Input on Finger Segments', *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct* p. 791–798.

[77] Utyansky, D. [2018], 'Digital Signal Processing for Frequency-Modulated Continuous Wave Radars'. `https://www.synopsys.com/dw/doc.php/wp/Digital_Signal_Processing_for_RADARs_CH.pdf`.

[78] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I. [2017], 'Attention is All You Need', *Proceedings of the 31st International Conference on Neural Information Processing Systems* p. 6000–6010.

[79] Velloso, E., Carter, M., Newn, J., Esteves, A., Clarke, C. and Gellersen, H. [2017], 'Motion Correlation: Selecting Objects by Matching Their Movement', *ACM Transactions on Computer-Human Interaction* **24**(3), 1–35.

[80] Venkatnarayan, R. H., Page, G. and Shahzad, M. [2018], 'Multi-User Gesture Recognition Using WiFi', *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services* pp. 401–413.

[81] Wachs, J., Stern, H., Edan, Y., Gillam, M., Feied, C., Smith, M. and Handler, J. [2007], 'Gestix: A Doctor-Computer Sterile Gesture Interface for Dynamic Environments', *Soft Computing in Industrial Applications* **39**, 30–39.

[82] Walter, R., Bailly, G. and Müller, J. [2013], 'StrikeAPose: Revealing Mid-Air Gestures on Public Displays', *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* p. 841–850.

[83] Wang, S., Song, J., Lien, J., Poupyrev, I. and Hilliges, O. [2016], 'Interacting with Soli: Exploring Fine-Grained Dynamic Gesture Recognition in the Radio-Frequency Spectrum', *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* p. 851–860.

[84] Wang, W., Liu, A. X. and Sun, K. [2016], 'Device-Free Gesture Tracking Using Acoustic Signals', *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking* p. 82–94.

[85] Wang, Y., Shu, Y., Jia, X., Zhou, M., Xie, L. and Guo, L. [2021], 'Multifeature Fusion-Based Hand Gesture Sensing and Recognition System', *IEEE Geoscience and Remote Sensing Letters* **19**, 1–5.

[86] Wang, Y., Wang, D., Fu, Y., Yao, D., Xie, L. and Zhou, M. [2022], 'Multi-Hand Gesture Recognition Using Automotive FMCW Radar Sensor', *Remote Sensing* **14**(10), 2374.

[87] Waugh, K., McGill, M. and Freeman, E. [2022], 'Push or Pinch? Exploring Slider Control Gestures for Touchless User Interfaces', *Nordic Human-Computer Interaction Conference* .

[88] Wen, H., Ramos Rojas, J. and Dey, A. K. [2016], 'Serendipity: Finger Gesture Recognition Using an Off-the-Shelf Smartwatch', *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* p. 3847–3851.

[89] Williamson, J. R. [2012], User Experience, Performance, and Social Acceptability: Usable Multimodal Mobile Interaction, PhD thesis, University of Glasgow.

[90] Williamson, J. R., Crossan, A. and Brewster, S. [2011], 'Multimodal Mobile Interactions: Usability Studies in Real World Settings', *Proceedings of the 13th International Conference on Multimodal Interfaces* p. 361–368.

[91] Wolf, K. [2016], Microgestures—Enabling Gesture Input with Busy Hands, *in* 'Peripheral Interaction: Challenges and Opportunities for HCI in the Periphery of Attention', Springer International Publishing, pp. 95–116.

[92] Wolf, K., Naumann, A., Rohs, M. and Müller, J. [2011], A Taxonomy of Microinteractions: Defining Microgestures Based on Ergonomic and Scenario-Dependent Requirements, *in* 'Proceedings of the 13th IFIP TC 13 International Conference on Human-Computer Interaction (INTERACT 2011)', Springer Berlin Heidelberg, pp. 559–575.

[93] Yang, Z. and Zheng, X. [2021], 'Hand Gesture Recognition Based on Trajectories Features and Computation-Efficient Reused LSTM Network', *IEEE Sensors Journal* **21**(15), 16945–16960.

[94] Yeo, H.-S., Flamich, G., Schrempf, P., Harris-Birtill, D. and Quigley, A. [2016], 'RadarCat: Radar Categorization for Input amp; Interaction', *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* p. 833–841.

[95] Yu, N., Wang, W., Liu, A. X. and Kong, L. [2018], 'QGesture: Quantifying Gesture Distance and Direction with WiFi Signals', *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* **2**(1).

[96] Zaiţi, I.-A., Pentiuc, -G. and Vatavu, R.-D. [2015], 'On Free-Hand TV Control: Experimental Results on User-Elicited Gestures with Leap Motion', *Personal and Ubiquitous Computing* **19**, 821–838.

[97] Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L. and Grundmann, M. [2020], 'MediaPipe Hands: On-device Real-time Hand Tracking', *arXiv 2006.10214* .

[98] Zhang, Z., Geiger, J., Pohjalainen, J., Mousa, A. E.-D., Jin, W. and Schuller, B. [2018], 'Deep Learning for Environmentally Robust Speech Recognition: An Overview of Recent Developments', *ACM Transactions on Intelligent Systems and Technology* **9**(5), 1–28.