Subtle Swipes: Radar-Based Micro-Gestures for Slider Control

Anith Manu Ravindran*
a.ravindran.1@research.gla.ac.uk
University of Glasgow
Glasgow, Scotland

Roderick Murray-Smith
University of Glasgow
Glasgow, Scotland
roderick.murray-smith@glasgow.ac.uk

Abstract

Subtle mid-air gestures, characterized by low-effort, discreet finger and hand movements, are a desirable approach to gestural interaction, as they reduce physical fatigue in frequent or prolonged use and improve social acceptability in many public and social contexts. Interaction using subtle micro-gestures is becoming increasingly feasible, enabled by recent advances in gesture sensing technologies. This study compares the use of macro-gestures and micro-gestures for slider control using mmWave radar, analyzing both objective performance and user preference. We investigate the effectiveness of these gestures for real-time slider control in interactive applications, utilizing mmWave radar for gesture sensing. We present a framework that integrates mmWave radar with a CNN-LSTM model for gesture recognition, enabling real-time interaction in two applications: a photo scroller to simulate discrete slider-based selection tasks and a video player to simulate continuous seeking interactions. A user study was conducted to evaluate subtle gestures and assess system performance based on task time, accuracy, and user experience scores. Results show that the developed system enabled fast and accurate control using subtle gestures, with participants reporting high user experience scores for the interaction techniques and applications. The findings highlight the practical potential of radar-based subtle gestures in ubiquitous computing contexts.

CCS Concepts

• **Human-centered computing** → **Gestural input**; **Empirical studies in interaction design**.

Keywords

mid-air gestures, radar sensing, slider control

ACM Reference Format:
Anith Manu Ravindran and Roderick Murray-Smith. 2025. Subtle Swipes: Radar-Based Micro-Gestures for Slider Control. In . ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Mid-air hand gestures 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 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.

Subtle gestures are non-intrusive and low-effort micro-movements utilizing small muscle groups, such as those in the fingers. They can help address common challenges in gestural interaction, such as *gorilla-arm effect*, where users experience discomfort and fatigue in the upper limbs due to repetitive gesturing [14]. Social acceptability is another critical factor that influences gestural interaction. Macro-gestures can draw unwanted attention, making users feel self-conscious or uncomfortable, particularly in public or shared spaces. This often limits the practical use of mid-air gestures in many public and social contexts [24]. Subtle gestures, with their minimal and discrete movements, provide a less conspicuous alternative, enabling interaction that is both effective and socially acceptable.

In the past, sensing subtle gestures heavily relied on wearable sensors such as force-sensitive resistors [9], surface electromyography [8], touch sensors [4], and ring-based sensors [2, 15]. Early non-wearable sensors, such as Wi-Fi [36, 1, 13, 27] and ultrasound [11, 29, 25, 5, 18], enabled mid-air gesture sensing without specialized hardware. While effective for detecting large hand motions and human posture, these methods lacked the spatial resolution needed for fine, low-amplitude movements.

Millimeter-wave (mmWave) radars have been gaining increasing attention in gesture interaction research due to their fine spatial resolution and high sensitivity to small movements [17, 28]. These properties make them particularly well-suited for capturing subtle micro-gestures. The application of mmWave radars in gesture recognition has been propelled by advancements in deep learning techniques. As a gesture is performed, the radar signals dynamically change, reflecting variations in movement patterns across different parts of the hand or body. Neural networks can effectively learn the temporal and spatial variations from radar signals, enabling them to classify gestures with high accuracies.

Current work in radar-based gesture recognition has focused on offline model performance evaluations. As a result, there is a gap in research involving real-time experiments that could uncover new interaction possibilities and guide the development of practical, user-friendly radar-based gesture interfaces. In this work, we explore the usability of radar-based subtle gesture recognition systems in real-time applications, particularly in slider-controlled interfaces. Sliders serve as a versatile interaction metaphor for a wide range of tasks, including navigating lists, adjusting numerical values, scrolling through menus, or exploring timelines. These interactions are fundamental to many everyday applications, such as media browsing, where users might scroll through options to select a video or song, or seeking through a video timeline requiring precise adjustments.

We present an experiment that evaluates slider control using radar-based subtle gestures, focusing on their performance and user experience compared to traditional macro-gestures. Two applications were developed: a *Photo Scroller* for discrete selection and a *Video Player* for continuous seeking. These applications utilize three candidate gestures: *Large Swipes*, *Thumb Swipes*, and *Pinch Swipes*, with the latter two representing *Subtle Swipes*. Participants performed tasks requiring either discrete selection or continuous seeking while the gestures were detected and recognized in real-time using a CNN-LSTM model.

The study assesses a range of performance metrics, including task time, time to target, recognition accuracy, error distance, overshoots, and undershoots, as well as user preferences and perceived usability through questionnaires and open-ended feedback. Results demonstrate that *Subtle Swipes* enable faster, more precise interactions with sliders, while also being rated higher in terms of comfort and social acceptability. These findings highlight the potential of radar-based subtle gestures to enhance user experiences in real-time slider-controlled applications.

2 Background

This section provides an overview of the foundational concepts and prior research that inform this study. It begins by explaining gesture recognition using mmWave radars, focusing on how radar data is processed and classified. The section then discusses subtle gestural interaction, highlighting its low-effort and discreet qualities.

2.1 Gesture Recognition Using mmWave Radars

Radar data is often represented as a *range-Doppler map* (RDM), which provides a detailed visualization of reflected energy intensity based on an object's distance and velocity. The RDM is constructed by processing radar responses through a 2-D fast Fourier transforms (FFT) that extract range and velocity information. This representation captures the dynamic motion patterns of gestures, such as variations in range and velocity over time.

Deep learning models have been extensively used for gesture classification from RDM sequences. Table 1 provides a summary of the different radar sensors, the types and numbers of gestures recognized, and the neural network architectures employed from various studies. 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). Dong et al. use a 3D-CNN architecture to recognize 16 macro-gestures [10]. 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 [7]. In addition, there are many studies [28, 12, 31] 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 [30], ResNet [20], and Transformer [6] 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.

2.2 Subtle Gestural Interaction

Pohl et al., in their paper "Charting Subtle Interaction in the HCI Literature", analyzed 55 HCI publications that used the term "subtle" [22]. 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: *non-intrusive feedback* (providing cues without disrupting focus), *low-effort input* (reducing physical exertion with small-scale gestures), *discreet interaction* (minimizing visibility for social acceptability), and *nudging users* (guiding behavior through subtle cues).

In this work, we primarily focus on the low-effort quality of subtle interactions using micro-gestures. Chan et al. [3] described micro-gestures as "detailed gestures in a small interaction space," emphasizing miniaturization for discreet input, while Wolf et al. [34, 33] 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 has demonstrated the potential of micro-gestures to reduce physical fatigue and improve usability. 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 [19]. Similarly, the *WristFlex* system uses minor hand movements, such as pinching two fingers [9]. The *Nenya* ring is another example which enables control through "small, discreet movements" [2]. The EMG controller by Costanza et al. detects subtle gestures from muscle contractions using surface electromyography, recognizing small hand movements through muscle activity signals [8]. 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 [4]. The low-effort nature not only helps reduce physical fatigue but also aligns with the concept of balancing between focused and casual interactions as outlined by Pohl and Murray-Smith [23]. The focused-casual continuum reflects how users adjust their engagement with technology based on context. While focused interactions demand heightened attention and greater effort, casual interactions are more laid-back, allowing users to remain partially engaged with their primary activities. Subtle gestures can support such casual interaction by minimizing cognitive and physical load.

Low-effort inputs are also linked to discreet interaction, as small, low-amplitude gestures are inherently less visible. Discreet interactions focus on hiding actions from others, particularly in social and public contexts where overt gestures might be considered distracting or socially awkward. This relationship is crucial for improving social acceptability. Williamson highlights the concept of everyday actions as performances, where individuals remain conscious of their surroundings and adapt their behavior based on social cues [32]. In such settings, large macro-gestures can make users feel self-conscious or uncomfortable, reducing their willingness to interact with devices. By contrast, subtle gestures have been shown to be

Table 1: Radar sensors, gesture types, and neural networks used in gesture recognition studies.

Radar	Gestures/Number	Classification Algorithm
IWR1642	Macro/9, Micro/1	3D-CNN [26]
AWR1642	Macro/16	3D-CNN [10]
Soli	Macro/10	LSTM [7]
IWR1443	Macro/6	LSTM [35]
Soli	Macro/7, Micro/4	CNN-LSTM [28]
Soli	Macro/4	CNN-LSTM [12]
AWR1642	Macro/8	CNN-LSTM [31]
AWR1642	Macro/6	VGG-16 [30]
AWR1843	Macro/3, Micro/3	2D + 3D ResNet18 [20]
Soli	Macro/20	Transformer [6]

more socially acceptable, as their discreet nature allows users to engage with technology without drawing the attention of others.

3 System Design

This section provides an overview of the proposed system used for real-time slider control with subtle gestures. The system design overview is shown in Figure 1. The process begins when a user performs a gesture within the detection range of the radar sensor. In this study, a Google Soli mmWave radar is used. The constant false alarm rate (CFAR) algorithm then detects and segments the gesture. The gesture segment is 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.

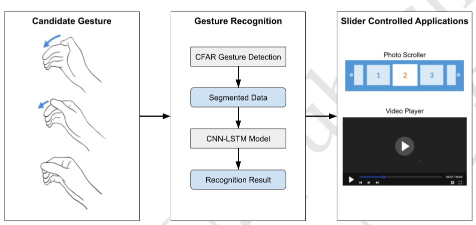

Figure 1: Overview of the system design for slider control using the Soli radar. Candidate gestures (*Large Swipes, Thumb Swipes, Pinch Swipes*) are performed within the Soli's detection range, detected by the CFAR algorithm, classified using a CNN-LSTM model, and then mapped to control actions in slider-based applications, such as a photo scroller or video player.

3.1 Candidate Gestures

Prior research on micro-gesture interaction—particularly in vision-based and capacitive sensing systems—has emphasized the usefulness of compact, repeatable sliding motions performed using just the thumb and index finger. For instance, Ultraleap's micro-gesture guidelines recommend gestures like short-range swipes and scrubs involving "only one or two fingers" to enable low-effort, precise input with proprioceptive and tactile feedback.[1] Similarly, the HandSense system recognized a vocabulary of thumb–index pinches and slides, designed for always-available input on AR headsets [21]. These gestures are well-suited to controlling continuous UI elements like sliders, as they can be performed in-place and repeated with minimal exertion. Building on these insights, this work adapts such micro-gesture principles—particularly the use of thumb–index sliding motions—for radar-based input. 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: , *Thumb Swipes*, *Pinch Swipes* and *Large Swipes*.

Pinch Swipes (Figure 2, middle) and *Thumb Swipes* (Figure 2, right) are collectively referred to as *Subtle Swipes*. *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. *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. 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.

Large Swipes (Figure 2, left) 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.

3.2 Gesture Detection and Recognition

In real-time gesture recognition, two components work together: *gesture detection* and *gesture classification*. The system first uses a CFAR algorithm to determine when a relevant motion occurs, and then a CNN-LSTM model classifies the detected motion into one of the target gestures.

The CFAR algorithm monitors the radar signal's intensity over time and dynamically adapts its threshold to filter out background

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

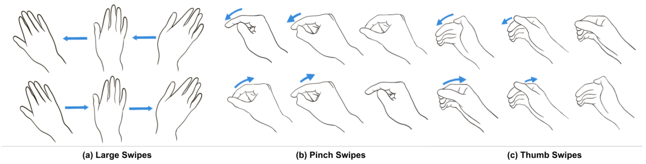

Figure 2: Candidate gestures for slider-based control using Soli. The figure illustrates three directional gestures: *Large Swipes*, *Pinch Swipes*, and *Thumb Swipes*.

noise. Whenever a series of consecutive frames exceeds this threshold, the system designates that interval as a gesture. In this work, a fixed window of 0.5s (13 RDM frames) is segmented and passed forward for classification.

The segmented RDMs are passed to a CNN-LSTM model for gesture classification (Figure 3). A hybrid CNN-LSTM model was chosen for its ability to capture both the spatial and temporal characteristics of gestures from RDM sequences. Convolutional layers learn high-level spatial features from each RDM frame (e.g., the shape or direction of a swipe), while the LSTM layers model the time-dependent evolution of these features across frames. We trained and evaluated this model using a newly collected dataset of gestures from eight participants, captured with a Google Soli radar configured for short-range sensing (up to 20cm). The dataset includes left and right *Thumb Swipes*, *Pinch Swipes*, and *Large Swipes*, systematically recorded from each participant. A leave-one-subject-out cross-validation yielded an average recognition accuracy of 95% (±0.3), demonstrating robust performance across different users.

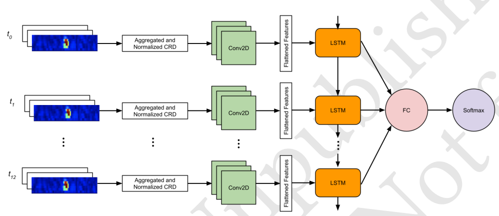

Figure 3: CNN-LSTM architecture used for subtle gesture recognition. Each RDM undergoes aggregation and normalization, followed by a 2D convolutional layer that extracts spatial features, which are then flattened and processed through an LSTM layer to capture temporal dependencies, culminating in a fully connected layer that feeds into a softmax layer for gesture classification.

3.3 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*.

3.3.1 Photo Scroller. This application (Figure 4, left) 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.

3.3.2 Video Player. This application (Figure 4, right) 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.

Control Modes. Two control modes are incorporated into the video player. The first mode involves the slider handle moving at a constant speed in the direction of the gesture, either left or right, until the user withdraws their hand from the Soli's sensing range, which signals the handle to stop. In the second control mode, the speed of the slider handle is dynamically adjusted based on the distance of the user's hand from the radar. The hand distance is scaled to control speed. As the hand moves closer to the sensor, the speed approaches zero. Conversely, as the hand moves farther away, the speed increases linearly, reaching its maximum at the radar's detectable limit of approximately 20cm.

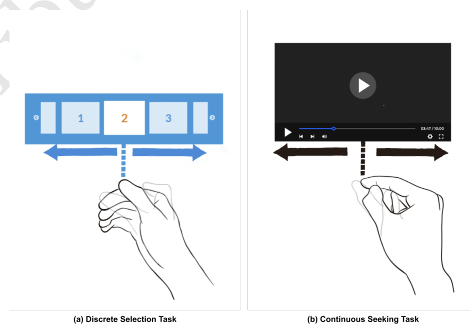

Figure 4: Applications developed for slider-based control using Soli. The system enables interaction with two main applications: a *Photo Scroller* for *Discrete Selection*, where users swipe to navigate left or right between fixed, discrete photo items, and a *Video Player* for *Continuous Seeking*, where users swipe to precisely seek through the video timeline.

4 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 type of system using radar-based subtle gestures provides a positive user experience for casual interactions.

4.1 Hypotheses

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 assumption 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*.

4.2 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*.

4.2.1 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.

4.2.2 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.

4.3 Procedure

The experimental setup is shown in Figure 5. 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 CNN-LSTM model was 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.

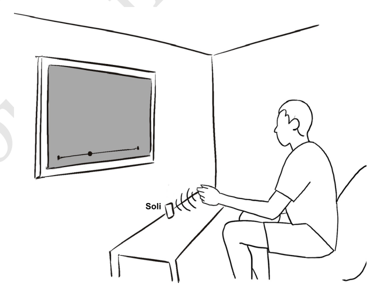

Figure 5: Experimental setup where participants were required to interact with the Soli radar system in a casual living room environment. Gestures performed within the detection range of the Soli sensor controlled the slider application displayed on the TV.

The order 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.

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 20cm of the sensor, the slider would turn green, signaling the start of the task and prompting the system to begin tracking and recognizing gestures. For *Large Swipes*, the task began as soon as the first swipe was recognized by the system.

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.

Following each task, participants were asked to complete a *User Experience Questionnaire* (UEQ) [16] to evaluate their experience with the two applications. The UEQ consists of 26 items divided into six scales. 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, designed to capture their feedback and preferences when comparing *Subtle Swipes* to *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.

4.4 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.

4.5 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 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.

5.1 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 6. 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 differences 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.

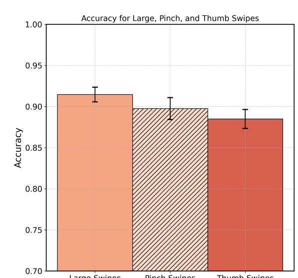

Figure 6: Recognition accuracy for t*Large, Pinch,* and *Thumb Swipes* during the *Discrete Selection* task. *Large Swipes* show the highest accuracy at 91.3%, while *Pinch Swipes* and *Thumb Swipes* have slightly lower accuracies at 89.4% and 88.2%, respectively. Error bars represent the standard error 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 (Figure 7, left) 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 (Figure 7, right) 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.

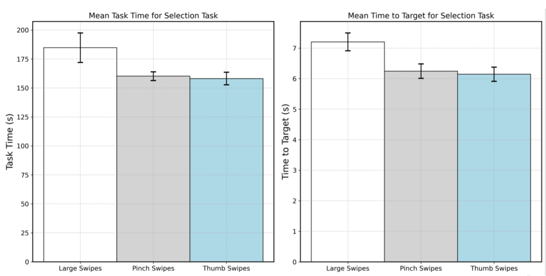

Figure 7: Mean task completion time (left) and time to target (right) for the *Discrete Selection* task using *Large Swipes*, *Pinch Swipes*, and *Thumb Swipes*.

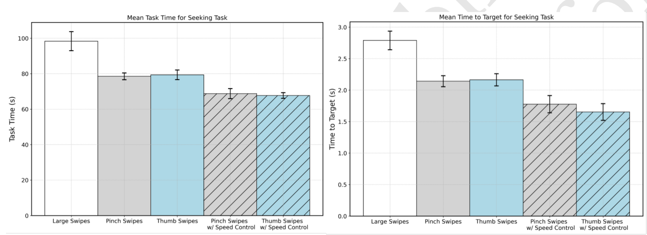

Figure 8: Mean task completion time (left) and time to target (right) for the *Continuous Seeking* task using *Large Swipes*, *Pinch Swipes*, and *Thumb Swipes* with and without *Speed Control*.

5.2 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*

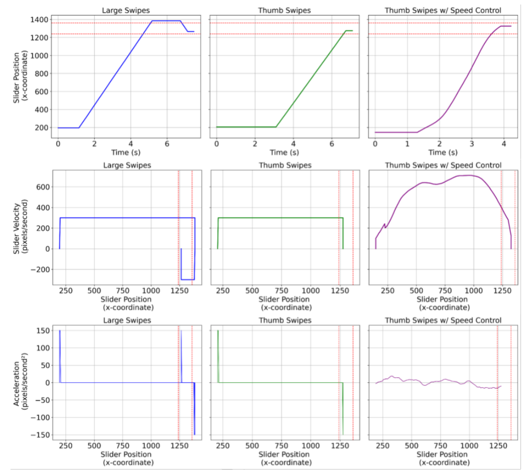

Figure 9: Slider dynamics for different gesture types and control modes during the *Continuous Seeking* task. The top row shows the slider position over time, highlighting distinct phases: initial reaction, ballistic motion, corrections near the target, and resting. The middle row presents phase-space plots of velocity versus position, where *Large Swipes*, and *Thumb Swipes*, exhibit instantaneous velocity changes, contrasting with the smooth bell-shaped velocity profile with *Speed Control*. The bottom row illustrates Hooke plots of acceleration versus position, emphasizing sharp accelerations for *Large Swipes* and *Thumb Swipes* swipes and gradual changes for *Speed Control*. The red dashed lines indicate the target area, and all data correspond to a single trial for target 7 with an index of difficulty of 4.

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 (Figure 8, left) 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 (Figure 8, right) using *Large 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 9 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*). The top row of the figure 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 The middle row of Figure 9 visualizes slider velocity against its position. For *Large Swipes* and *Thumb Swipes*, the slider accelerates instantaneously (as observed in the Hooke plots in the bottom row). It then moves towards the target at a constant velocity (300 pixels/second). Upon 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. Since users were able to achieve much higher speeds, the task time and time to target were significantly lower compared to the other control modes.

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 (Figure 10).

- **Overshoots:** 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.

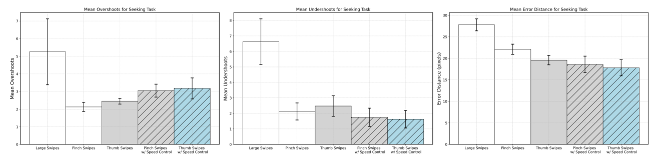

Figure 10: Comparison of performance metrics for the *Continuous Seeking* task across different swipe types. The left panel illustrates the mean overshoots, the middle panel shows the mean undershoots, and the right panel presents the mean error distance (in pixels). Results highlight significant improvements in all metrics when using *Pinch Swipes* and *Thumb Swipes* compared to *Large Swipes*, with further performance gains observed when *Speed Control* is applied.

- **Undershoots:** Mean 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 **??**. 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). Figure 11, display time series (top row) and phase space (bottom row) 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.

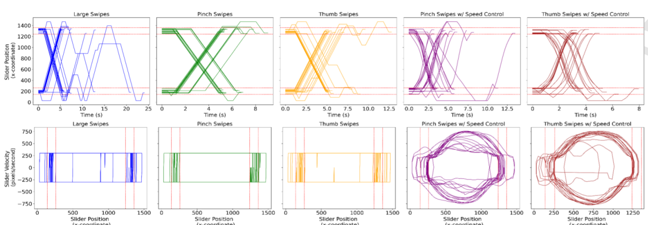

Figure 11: Slider dynamics for the Seeking Task for Large Swipes, Pinch Swipes, and Thumb Swipes (with and without Speed Control). The shown trials are for two targets with index of difficulty 4 for all participants. The top row illustrates time-series trajectories, showing *Pinch* and *Thumb Swipes*—both 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. The bottom row shows 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.

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 12. 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 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.

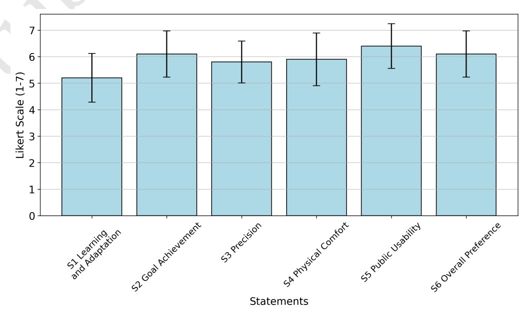

Figure 12: Mean Likert-scale ratings (1–7) for each of the six statements, with error bars indicating the standard deviation. Higher ratings reflect stronger agreement. Participants generally rated *Subtle Swipes* favorably, finding them easy to learn, efficient, precise, comfortable, publicly usable, and overall preferable compared to *Large Swipes*.

5.3 UEQ Results

Figure 13 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) [16]. The scores show that both applications performed positively across all scales. Figure 14 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."

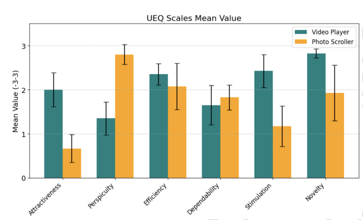

Figure 13: Comparison of mean scale scores between the Video Player and Photo Scroller applications.

- **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,

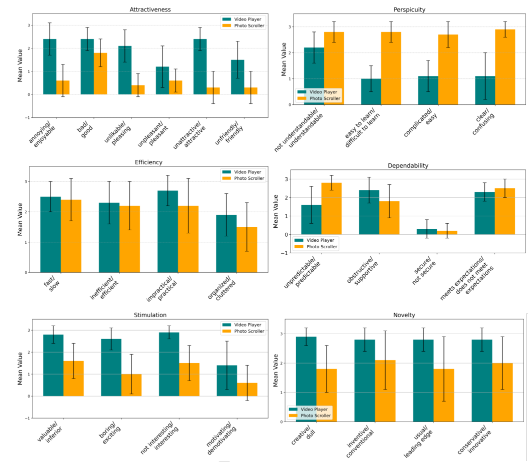

Figure 14: Mean value per item for each UEQ scale for Video Player and Photo Scroller applications.

"At first, the speed control was tricky to manage, and it felt unpredictable. But after a few 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."

6 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 12). 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**.

6.1 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, 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.

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.

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.

7 Conclusion

This study explored the potential of radar-based subtle gestures for slider control in interactive applications, focusing on their performance and user experience. We developed a framework for integrating mmWave radar gesture recognition into real-time slider-based applications, demonstrated through two use cases: a *Photo Scroller* for discrete selection and a *Video Player* for continuous seeking. The system utilized three candidate gestures: *Large Swipes*, *Thumb Swipes*, and *Pinch Swipes*, with the latter two representing *Subtle*

Swipes. A user study was conducted to evaluate the effectiveness of these gestures in terms of task performance, usability, and user preference. The results demonstrated that *Subtle Swipes* enabled faster and more precise interactions compared to *Large Swipes*. Participants completed tasks more quickly and with fewer errors when using *Subtle Swipes*. From a design perspective, the minimal physical exertion and reduced visibility of *Subtle Swipes* make them well-suited for interaction with devices in public spaces and social contexts. With the proliferation of mmWave radar sensing, subtle gestural interactions are poised to play an increasingly important role in the design of intuitive and user-friendly interfaces for a wide range of applications.

References

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

[2] Daniel Ashbrook, Patrick Baudisch, and Sean White. 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*, 2043–2046.

[3] Edwin Chan, Teddy Seyed, Wolfgang Stuerzlinger, Xing-Dong Yang, and Frank Maurer. 2016. User Elicitation on Single-hand Microgestures. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 3403–3414.

[4] Liwei Chan, Rong-Hao Liang, Ming-Chang Tsai, Kai-Yin Cheng, Chao-Huai Su, Mike Y. Chen, Wen-Huang Cheng, and Bing-Yu Chen. 2013. FingerPad: Private and Subtle Interaction Using Fingertips. *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, 255–260.

[5] Ke-Yu Chen, Daniel Ashbrook, Mayank Goel, Sung-Hyuck Lee, and Shwetak Patel. 2014. AirLink: Sharing Files Between Multiple Devices Using In-Air Gestures. *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 565–569.

[6] Jae-Woo Choi, Chan-Woo Park, and Jong-Hwan Kim. 2022. FMCW Radar-Based Real-Time Hand Gesture Recognition System Capable of Out-of-Distribution Detection. *IEEE Access*, 10, 87425–87434.

[7] Jae-Woo Choi, Si-Jung Ryu, and Jong-Hwan Kim. 2019. Short-Range Radar Based Real-Time Hand Gesture Recognition Using LSTM Encoder. *IEEE Access*, 7, 33610–33618.

[8] Enrico Costanza, Samuel Inverso, and Rebecca Allen. 2005. Toward Subtle Intimate Interfaces for Mobile Devices Using an EMG Controller. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 481–489.

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

[10] Xichao Dong, Zewei Zhao, Yupei Wang, Tao Zeng, Jianping Wang, and Yi Sui. 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.

[11] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. 2012. SoundWave: Using the Doppler Effect to Sense Gestures. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1911–1914.

[12] Eiji Hayashi, Jaime Lien, Nicholas Gillian, Leonardo Giusti, Dave Weber, Jin Yamanaka, Lauren Bedal, and Ivan Poupyrev. 2021. RadarNet: Efficient Gesture Recognition Technique Utilizing a Miniature Radar Sensor. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–14.

[13] Wenfeng He, Kaishun Wu, Yongpan Zou, and Zhong Ming. 2015. WiG: WiFi-Based Gesture Recognition System. *Proceedings of the 24th International Conference on Computer Communication and Networks*, 1–7.

[14] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-Air Interactions. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1063–1072.

[15] Lei Jing, Zixue Cheng, Yinghui Zhou, Junbo Wang, and Tongjun Huang. 2013. Magic Ring: A Self-Contained Gesture Input Device on Finger. *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*, 1–4.

[16] Bettina Laugwitz, Theo Held, and Martin Schrepp. 2008. Construction and Evaluation of a User Experience Questionnaire. In *HCI and Usability for Education and Work*. Springer Berlin Heidelberg, 63–76.

[17] Jaime Lien, Nicholas Gillian, M. Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. 2016. Soli: Ubiquitous Gesture Sensing with Millimeter Wave Radar. *ACM Transactions on Graphics*, 35, 4, 1–19.

[18] Kang Ling, Haipeng Dai, Yuntang Liu, Alex X. Liu, Wei Wang, and Qing Gu. 2022. UltraGesture: Fine-Grained Gesture Sensing and Recognition. *IEEE Transactions on Mobile Computing*, 2620–2636.

[19] Mingyu Liu, Mathieu Nancel, and Daniel Vogel. 2015. Gunslinger: Subtle Arms-down Mid-Air Interaction. *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, 63–71.

[20] Zhenyu Liu, Haoming Liu, and Chongrun Ma. 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.

[21] Viet Nguyen, Siddharth Rupavatharam, Luyang Liu, Richard Howard, and Marco Gruteser. 2019. HandSense: Capacitive Coupling-Based Dynamic, Micro Finger Gesture Recognition. *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, 285–297.

[22] Henning Pohl, Andreea Muresan, and Kasper Hornbæk. 2019. Charting Subtle Interaction in the HCI Literature. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–15.

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

[24] Julie Rico and Stephen Brewster. 2010. Usable gestures for mobile interfaces: evaluating social acceptability. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 887–896.

[25] Wenjie Ruan, Quan Z. Sheng, Lei Yang, Tao Gu, Peipei Xu, and Longfei Shangguan. 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*, 474–485.

[26] Xiangyu Shen, Haifeng Zheng, Xinxin Feng, and Jinsong Hu. 2022. ML-HGR-Net: A Meta-Learning Network for FMCW Radar Based Hand Gesture Recognition. *IEEE Sensors Journal*, 22, 11, 10808–10817.

[27] Raghav H. Venkatnarayan, Griffin Page, and Muhammad Shahzad. 2018. Multi-User Gesture Recognition Using WiFi. *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, 401–413.

[28] Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev, and Otmar Hilliges. 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*, 851–860.

[29] Wei Wang, Alex X. Liu, and Ke Sun. 2016. Device-Free Gesture Tracking Using Acoustic Signals. *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, 82–94.

[30] Yong Wang, Yuhong Shu, Xiuqian Jia, Mu Zhou, Liangbo Xie, and Lei Guo. 2021. Multifeature Fusion-Based Hand Gesture Sensing and Recognition System. *IEEE Geoscience and Remote Sensing Letters*, 19, 1–5.

[31] Yong Wang, Di Wang, Yunhai Fu, Dengke Yao, Liangbo Xie, and Mu Zhou. 2022. Multi-Hand Gesture Recognition Using Automotive FMCW Radar Sensor. *Remote Sensing*, 14, 10, 2374.

[32] Julie R. Williamson. 2012. *User Experience, Performance, and Social Acceptability: Usable Multimodal Mobile Interaction*. PhD Thesis. University of Glasgow.

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

[34] Katrin Wolf, Anja Naumann, Michael Rohs, and Jörg Müller. 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, 559–575.

[35] Zhaocheng Yang and Xinbo Zheng. 2021. Hand Gesture Recognition Based on Trajectories Features and Computation-Efficient Reused LSTM Network. *IEEE Sensors Journal*, 21, 15, 16945–16960.

[36] Nan Yu, Wei Wang, Alex X. Liu, and Lingtao Kong. 2018. QGesture: Quantifying Gesture Distance and Direction with WiFi Signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2, 1.