Wrist-Powered Touch: Evaluating Smartwatch-Based Touch Gesture Recognition for Interaction in Extended Reality

Pascal Knierim
University of Innsbruck
Innsbruck, Austria
pascal.knierim@uibk.ac.at

Jacob Höck
University of Innsbruck
Innsbruck, Austria
jacob.hoeck@student.uibk.ac.at

Thomas Kosch
HU Berlin
Berlin, Germany
thomas.kosch@hu-berlin.de

Abstract

The lack of tactile feedback and occlusion from visual tracking systems hinders touch interaction in Extended Reality (XR) environments. In this work, we present a method that enables touch gesture interaction on any physical surface using smartwatch-based inertial sensing. By using accelerometer data from a smartwatch, our approach captures micro-wrist movements to detect seven distinct touch gestures with 91.67% accuracy via a Long Short-Term Memory neural network. Our approach allows users to interact with XR interfaces anchored to everyday surfaces, such as tables or walls, while benefiting from natural haptic feedback. We introduce a dataset collected from 20 participants to demonstrate the feasibility through a controlled study. Our findings show that smartwatch sensing offers a low-cost, mobile, and accurate solution for extending XR input capabilities beyond the camera's view on physical surfaces, paving the way for more natural and privacy-preserving interaction in future XR systems.

CCS Concepts

• **Human-centered computing** → **Mixed / augmented reality**; **Interaction techniques**; *Ubiquitous and mobile devices.*

Keywords

Extended Reality, Interaction, Haptic Feedback, Touch, Smartwatch, Accelerometer, Dataset

ACM Reference Format:
Pascal Knierim, Jacob Höck, and Thomas Kosch. 2025. Wrist-Powered Touch: Evaluating Smartwatch-Based Touch Gesture Recognition for Interaction in Extended Reality. In *Mensch und Computer 2025 (MuC '25), August 31–September 03, 2025, Chemnitz, Germany*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3743049.3743090

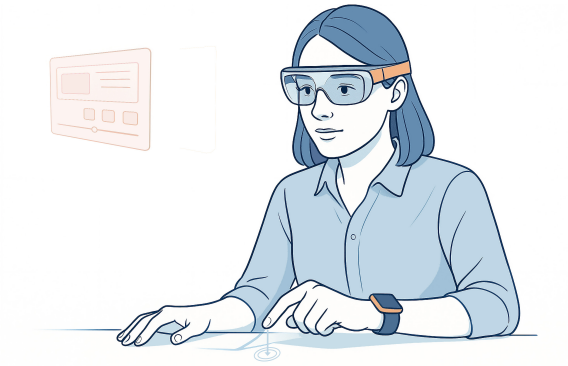

Figure 1: Leveraging accelerometer data from a smartwatch, this work investigates seven unique touch and swipe gestures, processed by a Long Short-Term Memory network, to enable natural interaction within extended reality environments.

1 Introduction

Extended Reality (XR) enables users to interact with digital content superimposed onto the real environment, creating immersive experiences [31]. While first-[33] and second-wave [1] XR head-mounted displays (HMDs) were expensive, bulky, uncomfortable, and limited with primarily targeting industry and research, XR has now reached the consumer market. Recent advances in sensor-based artificial intelligence facilitate processing inputs from advanced, always-on sensors, enabling seamless XR experiences [12]. At the same time, innovations in computing drive the improved usability of XR HMDs (e.g., Meta Orion[1]).

However, the convergence toward the form factor of regular glasses comes at a cost. Current XR glasses provide rich sensor input and immersive audio-visual output, but the push for lightweight, ergonomic smart glasses for all-day use comes at the cost of shorter battery life and reduced sensor capabilities. These trade-offs significantly impact the supported input modalities of lightweight and XR HMDs.

To overcome these limitations, we observe trends such as battery externalization[2] and sensor decoupling [17]. Following the concept of sensor decoupling, we propose utilizing modern smartwatches as external input sensors to enable accurate touch gesture detection. Smartwatches, equipped with energy-efficient inertial accelerometers, serve as a reliable data source for discreetly detecting wrist movements, enabling the recognition of touch interactions and gestures regardless of the surface they are performed on. Smartwatch-based sensing gained attention as a promising alternative, with prior studies using inertial data to recognize finger [18, 35] or on-surface gestures [5]. Nevertheless, these works neglect are limited to five gestures and XR-specific scenarios, where camera-based tracking is required to execute the gestures.

MuC '25, Chemnitz, Germany

ACM ISBN 979-8-4007-1582-2/25/08
https://doi.org/10.1145/3743049.3743090

[1] https://about.fb.com/news/2024/09/introducing-orion-our-first-true-augmented-reality-glasses
[2] https://www.apple.com/apple-vision-pro/specs

Our research fills this gap by exploring surface-based touch interaction in XR, removing the dependency on camera-based tracking [22, 36], and assessing the gesture detection effectivity of Long Short-Term Memory (LSTM)-based classification [5]. We envision scenarios where non-spatial applications, e.g., web browsers, calendars, or social media apps, are anchored to physical surfaces such as tables or walls. Yet, while common surfaces are not touch-sensitive, hand-tracking through vision-based approaches is negatively affected by the camera's field of view, self-occlusions, or lighting conditions [28].

We investigate how smartwatch-based accelerometer sensing can enable efficient touch-based gesture detection in XR environments. Specifically, we propose leveraging a smartwatch's inertial sensors to detect and classify gestures such as taps and swipes on physical surfaces [39], mimicking touchscreen-like interaction (see Figure 1). This approach eliminates error-prone detection near surfaces caused by self-occlusion, removes dependence on camera-based tracking, and supports optimized HMD hardware design. To validate our concept, we developed a gesture recognition system using an LSTM neural network trained on accelerometer data from 20 participants performing seven distinct gestures. The model achieves over 91% classification accuracy. Our findings show that smartwatches offer a low-cost, hands-free method for augmenting XR interaction with direct haptic feedback [34], expanding the input modality space for future systems. These findings indicate that regular smartwatches can serve as an effective, low-cost, hands-free augmentation for touch-based XR interaction. We conclude that integrating smartwatch-based input expands the range of input modalities for future XR systems. The overarching contribution of this paper is a method that enables smartwatch users to perform seven gestures on any surface using accelerometer data alone.

2 Related Work

The following section explains relevant previous work regarding traditional XR input techniques and sensing modalities.

2.1 Input Techniques

Traditional input methods such as keyboards, mice [27], and touchscreens are intuitive and widely used. Yet, these input methods are conceptualized for interaction in 2D spaces. Consequently, the input methods are not usable for XR, where 3D environments are more common. In this context, previous work explored various input techniques to facilitate the interaction in XR.

Hand-held controllers, a common commercially available input modality [21], are an industry standard. However, they occupy the user's hands, limiting the overall immersion and possible input space. As a consequence, previous work investigated using smartphones as a control device for XR interaction. Zhu and Grossman [38] presented a design space that guides designers on how augmented reality applications benefit from smartphone interaction and vice versa. Knierim et al. [13] evaluated the implementation of the design space using the smartphone as an interaction modality for augmented reality. Their results showed that using a physical touchscreen significantly outperformed mid-air gestures for interaction. This suggests that interaction in mobile AR environments can be enhanced by leveraging the smartphone as an always-available control device. Still, the envisioned interaction scenario requires users to have a smartphone available while offering limited space for haptic interaction.

In contrast, various researchers investigated hands-free interaction. Gaze-based [19, 26] input is hands-free, but either requires confirmation mechanisms or may fall risk to the Midas touch, reducing speed and precision. Gesture-based interaction is intuitive and accurate [16, 29], but can lead to fatigue over extended use [30], requiring careful design of the interactive elements to conform to the physical capabilities of humans [6]. Voice input is effective for high-level commands but lacks spatial precision and privacy protection and is challenged by noisy environments. Touch-based interaction offers a precise, familiar alternative that leverages XR's ability to anchor digital elements to physical surfaces with haptic feedback. However, current methods often depend on external cameras or wearables and struggle with detecting touch on arbitrary surfaces, limiting their suitability for lightweight, mobile XR systems. To address these limitations and explore more versatile input methods, researchers have investigated a range of sensing modalities aimed at enabling touch interaction without relying on bulky external hardware or constrained environments [24].

2.2 Sensing Modalities for Input

Several XR systems investigated vision-based touch input, leveraging depth cameras, computer vision, and bio-acoustic sensing. MirageTable utilizes a depth camera to enable freehand XR interaction on a projected tabletop [4]. Similarly, MRTouch detects touch interactions on flat surfaces in the field of view of the depth and infrared camera streams from Microsoft HoloLens [36]. EgoTouch utilizes an RGB camera usually built into XR headsets to track on-body touch interactions in real-time using deep learning [22]. In contrast to vision-based approaches, Skinput applies bio-acoustic sensing to detect skin taps via an armband with specialized sensors, enabling always-available, on-body input [9]. While these systems successfully detect touch on surfaces or the users' bodies, they introduce constraints such as a limited field of view, additional external hardware, or environmental dependencies.

Other approaches leverage wearable sensing technologies, such as IMUs and vibration-based sensing, to detect touch interactions. ViBand enhances smartwatch accelerometers by increasing sampling rates to 4 kHz, enabling micro-vibration sensing for touch interaction [15]. TapID employs dedicated wrist-worn inertial sensors to detect surface taps and finger identification for XR applications [20], enabling dictionary-supported typing on flat physical surfaces. Xu et al. showed that motion energy from a finger and wrist-worn from an inertial measurement unit can be used to discriminate different gestures with high accuracy, highlighting the versatility of a smartwatch for fine-grained gesture input [37]. Alternatively, Oh et al. [25] introduced a ring-mounted vibration-based system for identifying which fingers touch a surface.

Instead of requiring additional hardware, previous research investigated how the sensors of smartwatches, a wearable device that is unobtrusively worn on the wrist, can be used to interact in XR. Several surveys highlighted the capabilities of using smartwatches for implicit interaction. Gomes et al. [7] conducted a literature review, identifying a wide range of algorithms and sensors used for

gesture recognition in smartwatches. The authors find accelerometers and gyroscopes as the most prevalent sensors, and that LSTMs are among the key algorithms used. However, the review reveals that many studies focus on general or health-related applications, lacking investigations into spatial interaction and context-specific challenges such as those found in XR environments. Furthermore, Horbylon Nascimento et al. [11] consolidated smartwatch interaction methods that rely on gesture recognition, including applications from text input and gaming to robotic control. While they demonstrate the versatility of smartwatches in supporting hands-free, sensor-based interaction, they also point out a lack of consistency in evaluating interaction quality and contextual applicability.

In this context, Wen et al. [35] investigated how the accelerometer and gyroscope data can be used to distinguish between five fine motor finger gestures. Similarly, Li et al. [18] compared the classification accuracy for finger gestures using accelerometer, linear accelerometer, and gyroscope data between five classification techniques. The authors find that the classification is feasible for the five different classification techniques. However, both the work from Wen et al. [35] and Li et al. [18] focus on finger gestures in the air, missing the haptic component of surfaces. To this end, Chen et al. [5] developed a system that recognized on-surface hand gestures using only the accelerometer and gyroscope of an off-the-shelf smartwatch, enabling intuitive and fatigue-reducing interaction with nearby devices. The authors designed a gesture recognition model based on a 1D convolutional neural network and evaluated it through a user study with ten participants performing seven distinct gestures. Their findings show that the system achieves high accuracy, with F_1 scores of .992 in person-specific models, .878 across users, and .957 when using limited personal calibration data, demonstrating both effectiveness and generalizability. We distinguish our work by focusing on evaluating surface-based touch interaction within XR environments. This circumvents scenarios requiring camera-based tracking, which is limited by occlusion and field-of-view constraints. Furthermore, we evaluate an LSTM model (i.e., a neural network that is efficient for training a model with time series data) rather than a 1D convolutional neural network to classify gestures and evaluate performance on a larger, more diverse participant sample.

2.3 Summary and Research Question

Prior work explored a variety of XR input techniques, including hand-held controllers [21], keyboards [14], gaze-based [19, 26], gesture-based [16, 29], and voice-based interaction. While these approaches offer a promising interaction efficiency, they are affected by limitations such as fatigue [30], occlusion, or reliance on external hardware. Smartwatch-based sensing emerged as a promising alternative, with studies leveraging inertial data for finger [18, 35] or on-surface gesture recognition [5]. However, these research findings either overlook XR-specific use cases or rely on limited evaluation settings. Our work addresses this gap by focusing on surface-based touch interaction for XR, eliminating the need for camera-based tracking [22, 36], and evaluating an LSTM-based recognition system with a larger and more diverse participant group than prior efforts [5]. The gap in the literature motivates our research question:

RQ: To what extend can touch-based gestures on physical surfaces be classified using smartwatch accelerometer data?

3 System

The overall goal of this work is the development of reliable touch gesture recognition on arbitrary surfaces for XR environments using only the accelerometer data stream from a smartwatch. We aim to classify different tap and swipe gestures using an LSTM model. In the following, we describe the supported touch gesture interactions and LSTM model architecture and provide an overview of the hardware and software components used for the prototype.

3.1 Operationalized Tap and Swipe Gestures

The selection of tap and swipe gestures was based on existing gestures used on smartphones, their practicality, their epistemological rooting in related work [7, 11], and their relevance for interaction in XR environments. Single tap (ST) and double tap (DT) are fundamental interactions used when interacting with computing interfaces, making them intuitive for users. The prolonged tap (PT), adapted from the force tap, extends the functionality by enabling sustained interaction. Swinging gestures -up, down, left, and right (SU, SD, SL, SR), allow for directional input and are essential for navigating content in web browsers or social media applications.

3.2 LSTM Architecture

We decided to use an LSTM architecture to classify the touch events in combination with the smartwatch acceleration data. LSTMs retain information across the few-hundred-millisecond window in which each gesture unfolds, letting the model exploit temporal cues that a feed-forward or purely convolutional network would otherwise miss. We used PyTorch[3] to implement the neural network. Previous work showed that incorporating a dropout layer during training enhances the classification accuracy by reducing the likelihood of overfitting [10]. As a consequence, we decided to use a single-layer LSTM to increase the number of hidden units. We expected this to improve the model capacity. The final hidden state of the LSTM layer was extracted and fed into a fully connected layer to complete the classification architecture. We appended a Softmax layer, to sum up vectors into one. Figure 2 illustrates the model architecture.

3.3 Implementation

To implement the study prototype, we utilized a combination of wearable sensing and deep learning to classify the sensing data. We captured the wrist movement using accelerometer data from a smartwatch. The smartwatch streamed the accelerometer data with 500 Hz via Bluetooth low energy to a central processing unit (HP Gaming Pavilion). Then, the data was processed in real-time by segmenting the motion sequences to extract relevant features for classification (see Figure 3). The classification of touch gestures was performed using an LSTM neural network described above. The model was trained with data from 20 participants who performed pre-defined gestures that included single taps, double taps, prolonged taps, and directional swipes. The data was preprocessed

[3] https://pytorch.org

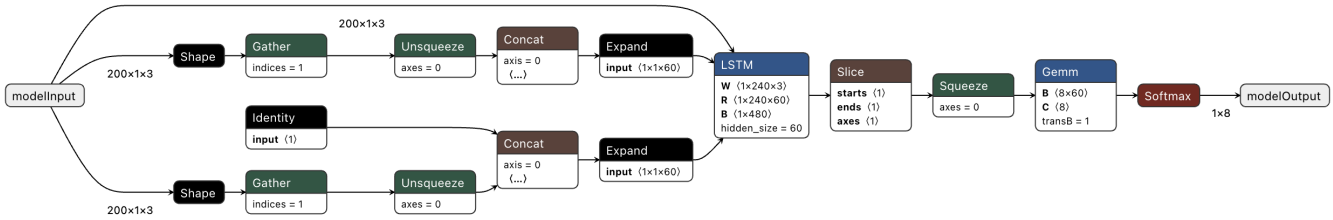

Figure 2: Architecture of the LSTM-based gesture classification model. The input is a time steps sequence with 3 features per frame. After reshaping and combining tensor dimensions, the sequence is processed by an LSTM layer with a hidden size of 60. The output of the final timestep is passed through a fully connected layer and softmax activation to produce an 8-class output.

by normalizing the accelerometer data. The training process included an Adam optimizer with a learning rate of .002, a batch size of 32, and a single-layer LSTM with 60 hidden units. The smartwatch accelerometer data was associated with the ground truth labels obtained from a smartphone-based annotation system (see Figure 3).

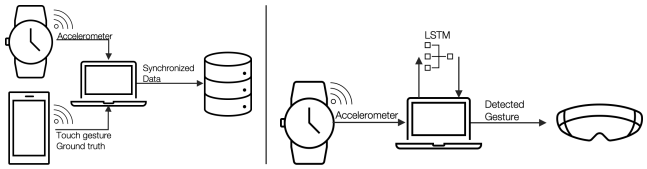

Figure 3: Illustration of the system architecture. Left: Overview of the architecture during gesture recording. Right: Overview of the system architecture during real-time gesture detection.

4 Dataset Collection

We evaluate the input efficiency of smartwatch-based touch gestures in a controlled lab study. In total, we collected accelerometer and touch data for 22,400 touch gestures from 20 participants. The dataset includes 3,200 individual recordings for each of the seven gestures, along with 3,200 additional no-interaction samples. It is freely available for further research[4]. In the following section, we describe the dataset creation in detail.

4.1 Participants

We recruited 20 participants aged between 16 and 87 years ($mean$ = 37.8, SD = 20, 62). Eight participants self-identified as female, while twelve participants self-identified as male. All participants were right-handed. Participants' body height ranged from 160 cm to 190 cm ($mean$ = 174.45 cm, SD = 10.36 cm). The majority of participants reported not wearing a smartwatch in their daily lives. We selected participants with varying technical expertise to generalize our findings to different demographic populations.

4.2 Procedure

After scheduling individual time slots with each participant and meeting in person, we began by explaining the overall course of the research and the objectives behind the collected data. We then provided an overview of the required tasks. Once the informed consent form was signed, participants completed a survey to gather general body data and demographics. Following this, we started the data collection process.

With the smartwatch worn on the right wrist and positioned in the initial resting pose, participants engaged in tapping and swiping actions as guided by the smartphone app. Each participant completed a sequence of seven different tap and swipe gestures, with 20 repetitions per gesture. A single run consisted of 140 taps and was completed for all four poses twice in a randomized order. Between runs, participants were allowed to take breaks as needed. Repeating all poses was introduced to encourage variation in positioning and reduce physical strain, eventually improving data diversity. Each pose (cf. Figure 4) was recorded separately, generating a total of eight data files per participant. On average, the data collection of 1,120 gestures (7 gestures x 20 repetitions x 4 poses x 2 trials) took 60 minutes. Each participant received a small reward (sweets) as a token of appreciation for participating in this extensive process.

4.3 Task

We asked our participants to perform a series of touch interactions, including taps and swipes while wearing a smartwatch on their right wrist. Participants completed these tasks across four different poses: sitting and standing with the smartphone display being horizontal, tilted at 45°, or vertically aligned. The goal was to capture variations in wrist movement to train a neural network. Participants followed instructions displayed on the smartphone, executing 1,120 taps and swipes. The participants were restricted from resting their right hand on a surface to ensure consistency and create a clean dataset for initial model training. This setup allowed us to collect a clean gesture data set that is representative of short-form interactions that are highly relevant in XR contexts.

4.4 Apparatus

The apparatus consists of a Samsung Galaxy A7 smartphone, a Bangle.js 2 smartwatch, an HP Gaming Pavilion (Intel Core i5 9300H, 2.4 GHz, 8 GB) notebook, and two angled smartphone mounts. The smartphone was used to display the target gesture to the user and record ground truth touch gesture data. Only text and an icon describing the current gesture were displayed to record unbiased interaction (see Figure 5). No lines to follow for swipes or timing cues for taps were shown. For different poses, the smartphone was

[4]https://github.com/pknierim/Wrist-Powered-Touch

Figure 4: Four different poses for data collection. (1) Sitting user interacting with a horizontal surface. (2) Sitting user interacting with a 45°angled surface. (3) Standing user interacting with a horizontal surface. (4) Standing user interacting with a vertical surface. The smartphone is used for ground truth recording.

placed on either a 90-degree or 45-degree stand. The Bangle.js 2 smartwatch, worn on the user's right wrist, streamed accelerometer data while performing instructed touch gestures. Both the ground truth touch data and accelerometer data were synchronized and stored using the HP Gaming Pavilion notebook.

Figure 5: A smartphone was employed to present gesture instructions to participants and simultaneously record the ground truth data for performed tap and swipe gestures.

4.5 Measures

We measured the three-axis accelerometer data from the Bangle.js 2 smartwatch worn on the participant's right wrist to detect and classify different types of touch gestures, including single taps, double taps, prolonged taps, and directional swipes (up, down, left, right). The smartwatch streamed raw sensor data at 500 Hz via Bluetooth low energy to a central program running on a PC, which processed the signals in real-time. Additionally, synchronized ground truth touch events were recorded using an Android smartphone. All the collected data was then used to train an LSTM neural network to classify different touch interactions based on wrist motion.

5 Model Evaluation

Using the created dataset, we trained various models to assess the feasibility of smartwatch-driven gesture recognition for XR. In the following sections, we report the classification performance of models capable of distinguishing different gesture combinations. Specifically, we evaluate a generalized model that considers all touch gestures (i.e., swipes and taps), as well as models trained separately on taps and swipes.

5.1 Data Processing and Training

Before training, the data is randomized to prevent class repetition from influencing the model. In each epoch, batches of data are converted into tensors and fed into the LSTM model, with loss calculated using mean square error. The Adam optimizer adjusted the model's internal parameters to minimize this loss. Training data accounts for 70% of all samples, while validation data is used periodically to track accuracy. If a lower validation loss is achieved, the model is saved. After 50 epochs, the model is tested on the remaining 15% of unseen data to assess final accuracy. The three models were trained for 200 epochs, using a batch size of 32 and a learning rate of .002, with a single-layer LSTM and 100 hidden units, a configuration that yielded the best accuracy.

5.2 Taps and Swipes

The first model was trained on all seven tap and swipe types: single tap, double tap, prolonged tap, up-swipe, down-swipe, left-swipe, and right-swipe, achieving an overall F_1 score of .916. The double tap and prolonged tap had the highest accuracy, both with an F_1 score of .94, while the single tap had the lowest accuracy at .89, followed by the down-swipe and right-swipe at .90. The confusion matrix (see Figure 6) indicates that single taps are sometimes misclassified as double taps, prolonged taps, or even right-swipes, whereas double and prolonged taps are rarely confused with other

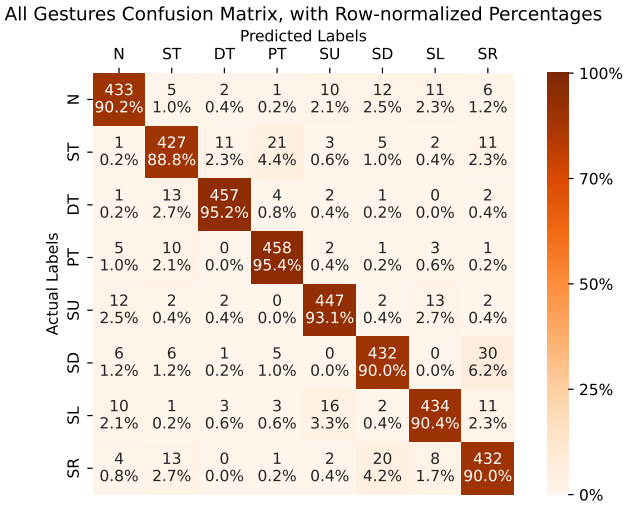

Figure 6: Confusion matrix of the final model, which classifies seven touch gestures (single tap (ST), double tap (DT), prolonged tap (PT), swipe-up (SU), swipe-down (SD), swipe-left (SL), and swipe-right (SR)) along with the none (N) category.

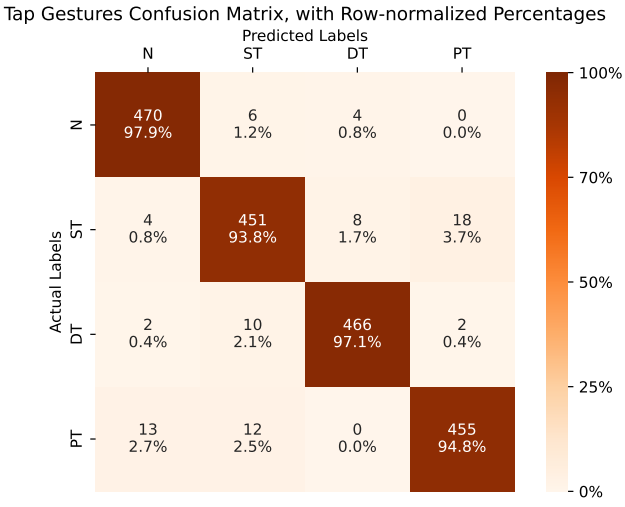

Figure 7: Confusion matrix of the final model, which classifies three tap gestures (single tap (ST), double tap (DT), prolonged tap (PT)) along with the none (N) category.

types. The model struggles most with distinguishing between down-swipes and right-swipes, as well as up-swipes and left-swipes, likely due to similar hand motions. Apart from these cases, swipes were classified with high accuracy. The training loss (see Figure 9 (orange)) approaches zero toward the end, suggesting that further training with the same input structure would not significantly improve performance.

5.3 Taps

This model, trained only on the three tap types, achieved the highest accuracy among all models, with an F_1 score of .959. Like the model trained on all interaction types, the single tap had the lowest accuracy at .94, often misclassified as a prolonged tap due to their similar input patterns. Double taps and prolonged taps were occasionally mistaken for single taps, as shown in Figure 7. However, the distinction between double and prolonged taps was the most precise, with almost no confusion between them. When comparing training loss progression, this model reached its plateau earlier than the model trained on all interaction types. While the all-types model stabilized at around 25,000 global steps, the tap-only model (see Figure 9 (blue)) achieved the same loss level at approximately 20,000 steps, indicating a faster convergence during training.

5.4 Swipes

The model trained only on swipes achieved a similar accuracy to the model trained on all interaction types, with an F_1 score of .916. The most accurately classified swipe was the up-swipe at .936, while the right-swipe had the lowest accuracy at .897, followed by the left-swipe at .914. The confusion matrix (see Figure 8) shows that down-swipes are most often misclassified as right-swipes, while up-swipes and left-swipes also tend to be confused with each other. The model performed best at distinguishing between up-swipes and down-swipes, as well as down-swipes and left-swipes, with almost no misclassification between these pairs. Similar to the all-types model, the none-type was sometimes mistaken for a swipe, but misclassifying a swipe as none was less frequent. While the training process was similar across all models, this swipe-only model took the longest to reach minimal training loss, requiring 30,000 global steps (see Figure 9 (gray)). However, it showed the fastest improvement during the initial training phases.

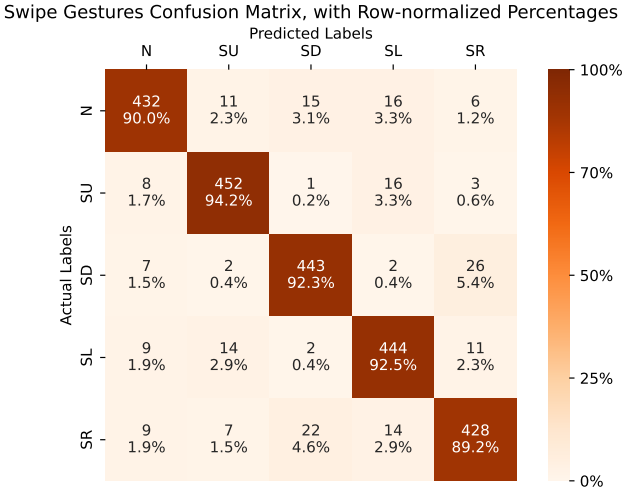

Figure 8: Confusion matrix of the final model, which classifies four swipe gestures (swipe-up (SU), swipe-down (SD), swipe-left (SL), and swipe-right (SR)) along with the none (N) category.

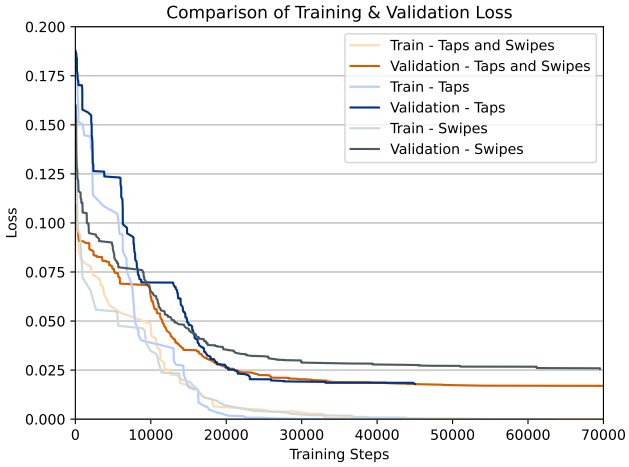

Figure 9: Minimum training and validation loss trajectories over training steps for the three evaluated models: Taps and Swipes (orange), Taps only (blue), and Swipes only (gray).

6 Discussion

We conducted a user study to generate a dataset of smartwatch acceleration data for seven distinct taps and swipe gestures. We trained and evaluated an LSTM model to utilize this data for accurate and efficient input detection in XR scenarios. The evaluation results indicate that incorporating smartwatch accelerometer data provides good recognition accuracy for simple touch inputs. In the following, we discuss the implications of our findings.

6.1 Performance Evaluation and Gesture-Specific Accuracy

Evaluation of the three trained models demonstrated consistently high accuracy. Depending on the gesture set, overall accuracy is above 91% for all models. The model trained exclusively on taps only achieved the highest accuracy with 95.94% (91.63% for swipes only and 91.67% for all seven gestures), confirming that distinct hand movements can be effectively recognized. Using LSTMs proved suitable for this classification task, eliminating the need for data augmentation.

Despite using a lower sampling rate of 500 Hz -compared to similar projects such as TapID [20], TapType [32], or ViBand [15], which employ refresh rates exceeding 1,000 Hz- the LSTM models still maintained high accuracy. The most challenging interaction types to distinguish were left-swipes and right-swipes. This was anticipated, given their lack of distinct wrist movement. When utilizing smartwatch-based touch interaction, we suggest either relying on other gestures or nudging the user, for example, visually, to perform extended swipe gestures over an extended physical distance.

In contrast, double taps and prolonged taps were classified with the highest accuracy, with double taps reaching an F_1 score of .973 in the tap-specialized model. This finding confirms current interaction concepts implemented in modern smartwatches [2]. Here, the user has the option to use the single gesture (excluding additional assistive options) of a double tap of index finger and thumb to navigate and confirm an action.

The swipes-only model performed well, but its accuracy was not significantly higher than that of the general model supporting all seven interactions, suggesting that taps were more straightforward to distinguish and contributed to overall performance. Thus, with the current implementation, we recommend relying on tap interactions when maximizing accuracy is critical. In scenarios where potential misclassifications are less disturbing but an extended interaction space is required, the general model represents an appropriate compromise.

6.2 Latency in Real-Time Classification

A critical limitation identified in this study is the latency inherent in our real-time classification prototype. The required accelerometer data window of approximately 500 ms, combined with transfer and processing times, results in a perceptible delay. Research indicates that users begin to notice delays when response times exceed 200 ms, which can disrupt the sense of direct manipulation in interactive applications [3]. For touch interfaces, this threshold is even lower, with studies showing that users prefer latencies as low as 10 ms [23]. This discrepancy can affect user experience, as higher latency leads to noticeable interaction delays.

To improve classification performance and reduce response times, we assume that streamlining the data processing workflow and decreasing the accelerometer data window, can further minimize latency, resulting in a more seamless user experience. Ultimately, we recommend considering minor delays when designing accelerometer-based interaction detection and developing a simplified user experience that maintains user engagement despite potential input delays.

6.3 Novel and Sleek Hardware Design for Privacy

Our prototype implementation and evaluation demonstrate that smartwatches can be facilitated to effectively detect touch and swipe interaction. In the context of mobile XR scenarios, this approach introduces a new interaction design space while addressing existing limitations. Notably, gestures no longer need to be performed within the field of view of tracking systems, such as the head-mounted visual or depth cameras.

By leveraging smartwatch-based interaction, sensors for input are decoupled from the HMD, reducing the dependence on multiple cameras. This, in turn, enables more compact and lightweight HMD designs while extending battery life. Additionally, interactions can occur without continuous environmental tracking, enhancing user privacy by minimizing persistent camera-based monitoring [8].

6.4 Limitations and Future Work

Our system demonstrates high classification accuracy for smartwatch-based gesture recognition, yet there remain exciting opportunities for further refinement and expansion.

Data Collection. The repetitive nature of the task posed a potential challenge to participant concentration and accuracy. However, this was effectively mitigated through strategic measures such as

randomizing tap types, incorporating frequent pose changes, and providing clear progress indicators.

Variations and Personalization. Variations in tap intensity were observed, presenting an interesting direction for further exploration. Some participants applied stronger taps, which enhanced event recognition, while others—particularly those accustomed to smartphone use—tended to tap more lightly, possibly due to subconscious associations with touchscreen interactions. Rather than intervening to standardize tap force, we embraced this variability to ensure the dataset remained representative of natural behavior. Future work could explore fine-tuning the model with individual user data. Adaptive and user-specific models that automatically adjust to individual movement styles and habits could further improve classification While the system already achieves high classification accuracy, these refinements will further improve efficiency, adaptability, and real-world integration, making it more effective for XR applications.

Gesture Set. We intentionally excluded further gestures like pinch or drag in this initial study to focus on gestures that can be reliably captured using a single smartwatch's accelerometer. More subtle gestures may require tracking fine-grained finger movement and, in many cases, coordination between multiple fingers or hands. These actions produce subtle motion signals that are difficult to distinguish from wrist-mounted inertial sensing alone, especially without additional sensors or higher-resolution input data, e.g., from finger-worn IMUs [24].

Our goal was to establish a feasible and reliable way of detecting a set of foundational, expressive touch gestures and swipes—that can be clearly identified using the available sensing modality. Future work can expand on this foundation by exploring multi-finger and compound gestures like pinch and drag, potentially by integrating additional sensor data or combining wrist sensing with finger-level instrumentation.

Integration with XR Frameworks. Our vision is to fully integrate our smartwatch-based approach into existing XR frameworks. While our method enables accurate and intuitive touch interactions, actual world performance in XR settings remains an open challenge. This requires optimized data synchronization between smartwatch input, classification, and XR rendering engine. By bridging this gap, smartwatch-based input can expand the XR interaction system, having the potential to become an alternative input method for short, haptic inputs on lightweight XR glasses.

7 Conclusion

In this work, we explored the classification effectivity of gestures on surfaces using accelerometer-based smartwatch sensing. We demonstrated that seven touch gestures performed on physical surfaces can be accurately classified using a Long Short-Term Memory (LSTM) neural network, achieving 91.67% accuracy (Macro-Averaged Precision: 94.88%; Macro-Averaged Recall: 93.02%). Our results indicate that smartwatch-based touch detection provides a viable, low-cost, and ergonomic alternative to traditional XR input methods, addressing common limitations such as occlusion and the lack of tactile feedback. Furthermore, our findings demonstrate the feasibility of integrating wearable sensors into XR as an interaction paradigm, providing a way for more natural and intuitive user experiences. Despite the strong classification performance, challenges remain in reducing latency for real-time applications. Future work should focus on optimizing processing efficiency and exploring user-specific adaptations to improve responsiveness and usability. By extending input modalities through wearable sensing, this work contributes to the broader vision of seamless, multimodal interaction in XR environments. The approach presented here serves as a reference implementation for further research into hybrid interaction techniques that blend traditional and emerging input methods, enabling meaningful and more accessible XR experiences through the use of touch input.

Acknowledgments

This work is partially supported by the German Research Foundation (DFG), CRC 1404: "FONDA: Foundations of Workflows for Large-Scale Scientific Data Analysis" (Project-ID 414984028).

References

[1] Christoph Anthes, Rubén Jesús García-Hernández, Markus Wiedemann, and Dieter Kranzlmüller. 2016. State of the art of virtual reality technology. In *2016 IEEE Aerospace Conference*. 1–19. doi:10.1109/AERO.2016.7500674

[2] Apple. October 25, 2023. Meta Quest Touch Pro Controllers. https://www.apple.com/newsroom/2023/10/apple-watch-double-tap-gesture-now-available-with-watchos-10-1/. Accessed: 2025-02-02.

[3] Ioannis Arapakis, Xiao Bai, and B. Barla Cambazoglu. 2014. Impact of response latency on user behavior in web search. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval* (Gold Coast, Queensland, Australia) *(SIGIR '14)*. Association for Computing Machinery, New York, NY, USA, 103–112. doi:10.1145/2600428.2609627

[4] Hrvoje Benko, Ricardo Jota, and Andrew Wilson. 2012. MirageTable: freehand interaction on a projected augmented reality tabletop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 199–208. doi:10.1145/2207676.2207704

[5] Junyu Chen, Hiroshi Saito, and Hiroshi Nakamura. 2023. Recognizing On-Surface Gesture Using Smartwatch. In *IEICE Conferences Archives*. The Institute of Electronics, Information and Communication Engineers.

[6] João Marcelo Evangelista Belo, Anna Maria Feit, Tiare Feuchtner, and Kaj Grønbæk. 2021. XRgonomics: Facilitating the Creation of Ergonomic 3D Interfaces. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 290, 11 pages. doi:10.1145/3411764.3445349

[7] Pedro Raphael Inácio Gomes, Murillo Santos de Castro, and Thamer Horbylon Nascimento. 2024. Gesture Recognition Methods Using Sensors Integrated into Smartwatches: Results of a Systematic Literature Review. In *Proceedings of the XXII Brazilian Symposium on Human Factors in Computing Systems* (Maceió, Brazil) *(IHC '23)*. Association for Computing Machinery, New York, NY, USA, Article 55, 11 pages. doi:10.1145/3638067.3638082

[8] Hilda Hadan, Derrick M. Wang, Lennart E. Nacke, and Leah Zhang-Kennedy. 2024. Privacy in Immersive Extended Reality: Exploring User Perceptions, Concerns, and Coping Strategies. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 784, 24 pages. doi:10.1145/3613904.3642104

[9] Chris Harrison, Desney Tan, and Dan Morris. 2010. Skinput: appropriating the body as an input surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 453–462. doi:10.1145/1753326.1753394

[10] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *ArXiv* abs/1207.0580 (2012). https://api.semanticscholar.org/CorpusID:14832074

[11] Thamer Horbylon Nascimento, Cristiane B. R. Ferreira, Wellington G. Rodrigues, and Fabrizzio Soares. 2020. Interaction with Smartwatches Using Gesture Recognition: A Systematic Literature Review. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*. 1661–1666. doi:10.1109/COMPSAC48688.2020.00-17

[12] Shahram Izadi. December 12, 2024. Android XR: The Gemini era comes to headsets and glasses. https://blog.google/products/android/android-xr/. Accessed: 2025-02-02.

[13] Pascal Knierim, Dimitri Hein, Albrecht Schmidt, and Thomas Kosch. 2021. The SmARtphone Controller: Leveraging Smartphones as Input and Output Modality for Improved Interaction within Mobile Augmented Reality Environments. *i-com* 20, 1 (2021), 49–61. doi:doi:10.1515/icom-2021-0003

[14] Pascal Knierim, Valentin Schwind, Anna Maria Feit, Florian Nieuwenhuizen, and Niels Henze. 2018. Physical Keyboards in Virtual Reality: Analysis of Typing Performance and Effects of Avatar Hands. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–9. doi:10.1145/3173574.3173919

[15] Gierad Laput, Robert Xiao, and Chris Harrison. 2016. ViBand: High-Fidelity Bio-Acoustic Sensing Using Commodity Smartwatch Accelerometers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. Association for Computing Machinery, New York, NY, USA, 321–333. doi:10.1145/2984511.2984582

[16] Seonho Lee and Junchul Chun. 2014. A stereo-vision approach for a natural 3D hand interaction with an AR object. *16th International Conference on Advanced Communication Technology* (2014), 315–321. https://api.semanticscholar.org/CorpusID:706320

[17] Kunjun Li, Manoj Gulati, Dhairya Shah, Steven Waskito, Shantanu Chakrabarty, and Ambuj Varshney. 2024. PixelGen: Rethinking Embedded Cameras for Mixed-Reality. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking* (Washington D.C., DC, USA) *(ACM MobiCom '24)*. Association for Computing Machinery, New York, NY, USA, 2128–2135. doi:10.1145/3636534.3696216

[18] Yande Li, Ning Yang, Lian Li, Li Liu, and Yi Yang. 2018. Finger gesture recognition using a smartwatch with integrated motion sensors. *Web Intelligence* 16, 2 (2018), 123–129. doi:10.3233/WEB-180378

[19] Mathias N. Lystbæk, Thorbjørn Mikkelsen, Roland Krisztandl, Eric J Gonzalez, Mar Gonzalez-Franco, Hans Gellersen, and Ken Pfeuffer. 2024. Hands-on, Hands-off: Gaze-Assisted Bimanual 3D Interaction. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 80, 12 pages. doi:10.1145/3654777.3676331

[20] Manuel Meier, Paul Streli, Andreas Fender, and Christian Holz. 2021. TapID: Rapid Touch Interaction in Virtual Reality using Wearable Sensing. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. 519–528. doi:10.1109/VR50410.2021.00076

[21] Meta. 2025. Meta Quest Touch Pro Controllers. https://www.meta.com/quest/accessories/quest-touch-pro-controllers-and-charging-dock/. Accessed: 2025-02-02.

[22] Vimal Mollyn and Chris Harrison. 2024. EgoTouch: On-Body Touch Input Using AR/VR Headset Cameras. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 69, 11 pages. doi:10.1145/3654777.3676455

[23] Albert Ng, Julian Lepinski, Daniel Wigdor, Steven Sanders, and Paul Dietz. 2012. Designing for low-latency direct-touch input. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) *(UIST '12)*. Association for Computing Machinery, New York, NY, USA, 453–464. doi:10.1145/2380116.2380174

[24] Ju Young Oh, Ji-Hyung Park, and Jung-Min Park. 2020. FingerTouch: Touch Interaction Using a Fingernail-Mounted Sensor on a Head-Mounted Display for Augmented Reality. *IEEE Access* 8 (2020), 101192–101208. doi:10.1109/ACCESS.2020.2997972

[25] Seungjae Oh, Chaeyong Park, Yo-Seb Jeon, and Seungmoon Choi. 2021. Identifying Contact Fingers on Touch Sensitive Surfaces by Ring-Based Vibratory Communication. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 208–222. doi:10.1145/3472749.3474745

[26] Hyung Min Park, Seok Han Lee, and Jong Soo Choi. 2008. Wearable augmented reality system using gaze interaction. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. 175–176. doi:10.1109/ISMAR.2008.4637353

[27] Edwin D. Reilly. 2003. Interactive system. https://api.semanticscholar.org/CorpusID:61951235

[28] Debajit Sarma and Manas Kamal Bhuyan. 2021. Methods, databases and recent advancement of vision-based hand gesture recognition for hci systems: A review. *SN Computer Science* 2, 6 (2021), 436.

[29] Daniel Schneider, Verena Biener, Alexander Otte, Travis Gesslein, Philipp Gagel, Cuauhtli Campos, Klen Čopič Pucihar, Matjazz Kljun, Eyal Ofek, Michel Pahud, Per Ola Kristensson, and Jens Grubert. 2021. Accuracy Evaluation of Touch Tasks in Commodity Virtual and Augmented Reality Head-Mounted Displays. In *Proceedings of the 2021 ACM Symposium on Spatial User Interaction* (Virtual Event, USA) *(SUI '21)*. Association for Computing Machinery, New York, NY, USA, Article 7, 11 pages. doi:10.1145/3485279.3485283

[30] Dominik Schön, Thomas Kosch, Florian Müller, Martin Schmitz, Sebastian Günther, Lukas Bommhardt, and Max Mühlhäuser. 2023. Tailor Twist: Assessing Rotational Mid-Air Interactions for Augmented Reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 400, 14 pages. doi:10.1145/3544548.3581461

[31] Maximilian Speicher, Brian D. Hall, and Michael Nebeling. 2019. What is Mixed Reality?. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–15. doi:10.1145/3290605.3300767

[32] Paul Streli, Jiaxi Jiang, Andreas Rene Fender, Manuel Meier, Hugo Romat, and Christian Holz. 2022. TapType: Ten-finger text entry on everyday surfaces via Bayesian inference. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 497, 16 pages. doi:10.1145/3491102.3501878

[33] Ivan E Sutherland et al. 1965. The ultimate display. In *Proceedings of the IFIP Congress*, Vol. 2. New York, 506–508.

[34] Peng Wang, Xiaoliang Bai, Mark Billinghurst, Shusheng Zhang, Dechuan Han, Mengmeng Sun, Zhuo Wang, Hao Lv, and Shu Han. 2020. Haptic Feedback Helps Me? A VR-SAR Remote Collaborative System with Tangible Interaction. *International Journal of Human-Computer Interaction* 36, 13 (2020), 1242–1257. doi:10.1080/10447318.2020.1732140

[35] Hongyi Wen, Julian Ramos Rojas, and Anind K. Dey. 2016. Serendipity: Finger Gesture Recognition using an Off-the-Shelf Smartwatch. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 3847–3851. doi:10.1145/2858036.2858466

[36] Robert Xiao, Julia Schwarz, Nick Throm, Andrew D. Wilson, and Hrvoje Benko. 2018. MRTouch: Adding Touch Input to Head-Mounted Mixed Reality. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (2018), 1653–1660. doi:10.1109/TVCG.2018.2794222

[37] Chao Xu, Parth H. Pathak, and Prasant Mohapatra. 2015. Finger-writing with Smartwatch: A Case for Finger and Hand Gesture Recognition using Smartwatch. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications* (Santa Fe, New Mexico, USA) *(HotMobile '15)*. Association for Computing Machinery, New York, NY, USA, 9–14. doi:10.1145/2699343.2699350

[38] Fengyuan Zhu and Tovi Grossman. 2020. BISHARE: Exploring Bidirectional Interactions Between Smartphones and Head-Mounted Augmented Reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. doi:10.1145/3313831.3376233

[39] Peide Zhu, Hao Zhou, Shumin Cao, Panlong Yang, and Shuangshuang Xue. 2018. Control with Gestures: A Hand Gesture Recognition System Using Off-the-Shelf Smartwatch. In *2018 4th International Conference on Big Data Computing and Communications (BIGCOM)*. 72–77. doi:10.1109/BIGCOM.2018.00018