Mudra: Enabling Microgesture Recognition on COTS Smartwatches

Sejal Bhalla*
University of Toronto
Toronto, Ontario, Canada
sejal@cs.toronto.edu

Benson Chou
University of Toronto
Toronto, Ontario, Canada
benson.chou@mail.utoronto.ca

Tianyu Xu
Google
Mountain View, USA
cadyxu@google.com

Karan Ahuja
Google
Seattle, USA
kahuja@northwestern.edu

Alex Mariakakis
University of Toronto
Toronto, Ontario, Canada
mariakakis@s.toronto.edu

Ishan Chatterjee
Google
Seattle, USA
ishanc@google.com

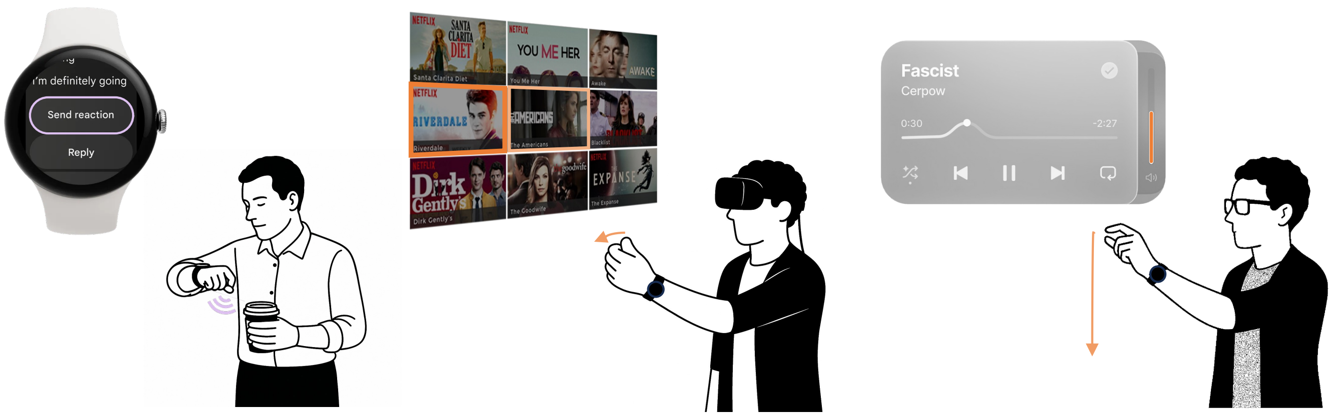


Figure 1: Mudra uses only IMU sensors on a commodity smartwatch to provide microgesture control for smartwatches, media players, and mixed reality interfaces. It supports microgestures including (left) Finger Tap, (middle) Swipe, and (right) Pinch-and-Drag. The latter two gestures are directional, allowing for flexible UI navigation.

Abstract

Subtle finger movements, or "microgestures", enable low-fatigue and discreet interactions. However, prior work often relies on custom hardware or recognizes isolated gestures, with limited exploration of directional input on commodity devices. We introduce Mudra, a microgesture recognition system that leverages inertial sensors in commercial off-the-shelf (COTS) smartwatches to support a broad range of interactions. Mudra recognizes four primary microgestures: Finger Tap and Surface Tap for selection, and Swipe and Pinch-and-Drag for targeting. Through a user study with 30 participants, we demonstrate that Mudra generalizes effectively across users, achieving an average microgesture classification accuracy of 0.93 ± 0.04. Incorporating directional variants of Swipe and Pinch-and-Drag, we design and evaluate gesture sets for 1D and 2D navigation, achieving user-independent accuracies of 0.93 ± 0.06 and 0.86 ± 0.07, respectively. By enabling subtle, directional input on commodity devices, Mudra paves the way for complete gesture control for smartwatches, media players, and extended reality interfaces.

CCS Concepts

• **Human-centered computing → Gestural input**; *Ubiquitous and mobile computing*.

Keywords

Microgestures, smartwatch, IMU, tapping, swiping

ACM Reference Format:
Sejal Bhalla, Benson Chou, Tianyu Xu, Karan Ahuja, Alex Mariakakis, and Ishan Chatterjee. 2018. Mudra: Enabling Microgesture Recognition on COTS Smartwatches. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (GI '26).* ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Mid-air gestures performed with arms and hands offer a natural and unconstrained method for interacting with digital systems.

*Work done during an internship at Google.

GI '26, Kitchener-Waterloo, Canada

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

However, executing them can lead to fatigue and discomfort, particularly when performed over extended durations [10, 26]. Microgestures — subtle finger movements that require minimal effort — present a compelling alternative. By relying on small movements of the thumb and finger, microgestures align well with familiar touchscreen interactions [4]. They are less conspicuous compared to overt arm or hand motions [5], making them more acceptable for everyday use. Microgestures also accommodate situations when users only have one hand available (e.g., due to encumbrance [14, 33, 37, 38], performing an auxiliary task [7, 20], or physical impairments [44]). Overall, these characteristics make microgestures particularly valuable for quick and discreet interactions such as control of smartwatches, media players, or extended reality (XR) interfaces [15, 19].

Despite their advantages, sensing microgestures remains difficult. Most gesture recognition systems prioritize wrist motions [23, 27, 51] and hand waves [22, 57] because they create obvious movements that can be detected at the forearm or wrist; microgestures do not produce such noticeable motions. Many efforts to address this challenge have relied on form factors such as rings [6, 9, 34], gloves [48], cameras embedded in VR headsets [19], and even fingernail sensors [17]. While these systems can achieve high accuracy, their reliance on custom hardware and potential privacy concerns preclude their practical adoption. Others have proposed using wrist-worn devices for microgesture recognition, often augmenting them with additional sensors to improve sensitivity to fine motor movements [18, 50]. Those that support swiping gestures are typically restricted to unidirectional motions at a single speed [47, 51]. Recognizing diverse and directional microgestures using unmodified hardware would significantly broaden access to microgesture control and their applicability toward more complex UI navigation.

We propose Mudra, a robust microgesture recognition system that leverages the motion sensors embedded in commodity smartwatches. Guided by gesture elicitation studies from prior work [3, 4, 14, 24], we focus on two core input primitives: *selection* via taps (Finger Tap, Surface Tap) and *targeting* via directional motions (Swipe, Pinch-and-Drag). Pairing these gestures enables 1D and 2D navigation on unmodified devices while keeping the complete set of gestures compact. Although these gestures are intuitive to describe, their execution varies significantly across users, making recognition difficult using simple heuristics. We adopt a machine learning approach to overcome this challenge, employing residual convolutional neural networks to analyze 6-axis IMU data captured from an unmodified smartwatch. Since slower gestures generate weaker motions, we use speed classification as a gating mechanism to trigger direction classification and reduce false positives.

We evaluated Mudra through a user study with 30 participants. Using leave-one-subject-out cross-validation, we show that our system achieves an average user-independent accuracy of 0.93 ± 0.04 in classifying between four microgesture categories. When we incorporate directional variants of those gestures for 1D and 2D UI navigation, Mudra achieves an average accuracy of 0.93 ± 0.06 and 0.86 ± 0.07, respectively.

To summarize, our paper makes the following contributions:

(1) Mudra, a system that relies on motion data captured using commodity smartwatches to recognize both selection and targeting microgestures, enabling gesture sets for 1D and 2D UI navigation,
(2) A comprehensive evaluation of Mudra's performance across 30 participants, demonstrating high performance across the 1D and 2D gesture sets,
(3) A hierarchical classification approach combined with a gating mechanism for robust recognition of directional gestures, addressing challenges introduced by variations in gesture speed.

2 Related Work

For our overview of related work, we first discuss microgesture sets that have been proposed or elicited from potential users. We then review related wearable systems, with a particular focus on wrist-worn devices designed for microgesture detection.

2.1 Microgestures

The term "microgestures" is broadly used to describe one-handed motions that engage some combination of the wrist, thumb, and finger. Microgestures can be used to provide less strenuous forms of input since they require less movement than mid-air macrogestures [10], and they can also be used in situations where a person only has one hand available [7, 14, 20, 33, 37, 38]. For device input specifically, the two most important capabilities are **selection** and **targeting** [11, 25, 32]. Targeting involves navigating to a desired object or interface element, which may include scrolling through content, switching between applications, or returning to a previous screen. Selection, on the other hand, typically involves confirming a choice, activating a function, or opening an application. In many cases, selection commands can also serve as toggles when consecutive selection inputs trigger the opposite action—for example, using a single tap gesture to play or pause media, or to mute and unmute audio.

The design space of single-handed microgestures has been extensively explored [3, 4, 14, 24], with prior work in HCI establishing taxonomies informed by gesture elicitation studies and literature surveys. Li et al. [24] proposed a set of 33 microgestures designed by certified ergonomists as an alternative to handheld controllers for VR and AR interactions. Forty users evaluated these gestures, linking them to 20 common AR/VR commands based on usability, comfort, and preference. Similarly, Chan et al. [4] conducted a study where participants performed microgestures with their palms and fingers to execute commands for applications like music playback and text editing. Both studies revealed a strong preference for tapping and swiping gestures, largely due to users' familiarity with touchscreen interactions.

When considering microgestures, taps provide an intuitive choice for selection, while multidirectional swipes and dragging motions naturally lend themselves to targeting tasks. Prätorius et al. [36] and others have mapped out the design space of thumb-to-finger microgestures, detailing possible swipe interactions across the radial, volar, ulnar, and dorsal sides of the fingers. However, most gesture recognition systems treat swipes as unidirectional motions [22, 47, 51], typically performed with the

Table 1: A summary of state-of-the-art microgesture recognition systems. The systems are coded according to their sensing modality, gesture set, suitability for commercial off-the-shelf (COTS) devices, and support of directional swiping motions.

System	Year	Form Factor	Modality	Gesture Set	COTS	Direction Support
DigiTouch [48]	2017	Glove	Pressure	Tap, Force Tap, Sliding, Chording	✗	✓
SparseIMU [40]	2023	Glove	IMU	Tap, Finger Flexion / Extension, Finger Abduction / Adduction, Finger Circumduction	✗	✓
EFRing [6]	2023	Ring	Electric Field	Tap, Double Tap, Thumb Swipe Up / Down / Left / Right	✗	✓
AuraRing [34]	2019	Ring	Magnetic Tracking	Hand Pose	✗	✓
OptiRing [46]	2023	Ring	Optical	Stateful Pinch, Thumb Swipe Left / Right, Continuous 1D Input	✗	✓
Tomo [57]	2015	Wrist-worn	EMG	Finger Pinch, Fist, Stretch, Wave Left / Right, Thumbs Up, Spider-Man	✗	✗
EM-Press [27]	2016	Wrist-worn	EMG	Finger Rotation, Wrist Flexion / Extension, Wrist Abduction / Adduction, Fist, Point Finger	✗	✗
EchoWrist [23]	2024	Wrist-worn	Acoustic	American Sign Language, Wrist Flexion / Extension, Wrist Ulnar Deviation, Finger Flexion	✗	✗
VibAware [18]	2023	Wrist-worn	Acoustic	Tap, Swipe	✗	✗
AO-Finger [50]	2023	Wrist-worn	Acoustic, Optical	Flick, Pinch, Tap, Thumb Swipe Left / Right	✗	✓
ViBand [22]	2016	Wrist-worn	IMU	Two-handed Gestures, On-body Touch Input, Pinch, Flick, Snap, Wave Up / Down, One Finger Rub	✓	✗
Serendipity [47]	2016	Wrist-worn	IMU	Pinch, Tap, Rub Fingers, Squeeze, Wave	✓	✗
Agarwal et al. [1]	2017	Wrist-worn	IMU	Fist, Flick, Pinch, Tap, Thumb Swipe, Palm Swipe Forward / Backward	✓	✗
Xu et al. [51]	2022	Wrist-worn	IMU	Clench, Double clench, Pinch, Double Pinch, Peace, Slide, Wrist Rotate In / Out, Wrist Deviate In / Out, Wrist Flexion / Extension	✓	✗
Mudra	**2026**	**Wrist-worn**	**IMU**	**Finger Tap, Surface Tap, Thumb Swipe Left / Right, Thumb Swipe Up/Down, Pinch-and-Drag Up / Down**	✓	✓

thumb on the radial or volar side of the index finger. Exceptions include STMG [19] and SparseIMU [40], which use a camera-instrumented headset and an IMU-instrumented glove for swipe direction classification. Mudra aims to enable selection and multidirectional targeting to facilitate more nuanced and expressive interactions, all while relying only on commodity sensors. These capabilities unlock convenient targeting for common UIs such as smartwatch menus, media playback on earbuds, or smartglass head-up displays [29].

2.2 Microgesture Recognition

Table 1 summarizes recent systems designed for microgesture recognition. We review prior work in this space based on the form factor and sensing principle of the hardware.

2.2.1 Non-wrist-worn Systems. Since microgestures are subtle, the most straightforward way of detecting them is by placing sensors along the fingers. Gloves and rings provide this affordance in an inconspicuous way. For example, Miller et al. [31] developed a glove that had conductive threads woven perpendicularly and orthogonally along the fingers. Beyond supporting discrete microgestures, users could put their fingers together to create 2D trackpad-like surfaces for more continuous input. Systems like DigiTouch [48], Argot [35], and KITTY [21] offered similar affordances with conductive elements embedded in a glove's fabric; however, these systems require large interaction areas since they were primarily designed for thumb-to-finger text entry.

To provide a less conspicuous surface for gestures and swiping, Yoon et al. [53, 54] proposed multiple finger-worn systems resembling a sleeve on a single finger. Their systems were able to detect not just contact but also pressure and strain due to finger bending, providing additional dimensions for microgesture expressions. Miniaturizing the interaction surface even further, the NailO system by Kao et al. [17] explored the potential of a fingernail-mounted interaction surface using printed electrodes.

Ring form factors offer a more convenient option for microgesture recognition. These devices first became popular for tap and swipe interactions comparable to a trackpad, using modalities ranging from motion [8, 49, 55] and audio [56] to even optical data [46, 52]. More recently, systems like AuraRing [34] and EFRing [6] have utilized electromagnetic signals to deliver more continuous forms of input.

2.2.2 Wrist-worn Systems. Wrist-worn devices are attracting increasing interest as a platform for microgesture recognition due to their superior affordances and widespread adoption. Many prominent works have instrumented existing wristbands with additional sensors for this task. Tomo [57] relied on electromyography (EMG) to recognize the extension and flexion of fingers, while EMPress [27] combined EMG with pressure sensing for both finger and wrist gestures. Beamband [13] and EchoWrist [23] utilized ultrasonic sensors on the wrist to detect hand poses and infer hand gestures. Similarly, VibAware [18] and AO-Finger [50] prototyped wristbands with custom acoustic and optical sensors, respectively, to detect subtle taps and swipes in various contexts.

Leveraging motion sensors in commodity smartwatches for microgesture recognition presents notable challenges due to their placement away from the fingers. Nevertheless, some systems have demonstrated promising performance. Serendipity by Wen et al. [47] recognized five gestures — Pinch, Tap, Rub, Squeeze, and Wave — achieving an average F1 score of 0.87 using user-dependent machine learning models. A similar system by Agarwal et al. [1] recognized seven gestures: Flick, Squeeze, Pinch, Tap, Non-Directional Thumb Rubbing, Palm Flexion, and Palm Extension. Using a subject-agnostic support vector machine adapted with a few subject-specific gesture instances for calibration, they were able to achieve a classification accuracy of 0.92. While both systems offered high classification performance and moderate gesture expressivity, their reliance on user calibration reduced their suitability for out-of-the-box use. Moreover, the absence of directional input limited their applicability for navigation-based interactions. Xu et al. [51] examined directional inputs using a gesture customization approach that required minimal exemplars to achieve robust performance. Their system, trained on 6-axis IMU data from smartwatches, performed well on coarse gestures such as wrist turns, rotations, and clenching. However, recognition performance dropped for subtle gestures, with non-directional thumb swipes yielding an F1 score of only 0.70.

Mudra belongs to the latter category of wrist-worn microgesture recognition systems, relying solely on the hardware available in unmodified commodity smartwatches. Unlike prior work, it enables directional input through subtle finger movements without requiring exaggerated gestures such as hand waves or wrist rotations. To the best of our knowledge, no existing system concurrently satisfies both objectives: enabling subtle, directional input on unmodified smartwatches and generalizing across users without requiring personalized calibration.

3 Mudra System Design

In this section, we first introduce the primary gestures and gesture sets supported by Mudra, followed by a description of its sensing principle and an overview of the system architecture and data processing pipeline.

3.1 Gesture Sets

Mudra is designed around a set of microgestures that support selection and targeting as two key primitives for 1D and 2D UI navigation. 1D navigation is well-suited for linear content traversal, such as scrolling through image galleries or discussion threads. 2D navigation enables control over interfaces with orthogonal commands such as music track selection (previous/next) and volume adjustment (up/down) in a media player. We investigate four primary gestures across two categories:

Selection Gestures

- **Finger Tap** and **Surface Tap** gestures offer quick, discrete input semantically mapped to selection or toggle commands [4, 24]. Finger Tap involves a brief contact between the thumb and index finger, while Surface Tap is performed when the index finger taps on a surface, often when the hand is partially occupied. Together, these gestures cover a range of interaction scenarios, supporting both hands-free and engaged use cases.

Targeting Gestures

- **Swipe** gestures leverage the natural movement of the thumb along distinct anatomical surfaces [36]. For instance, radial swipes can be performed by sliding the thumb along the curved edge of the index finger, evoking the tactile sensation of turning a physical knob. In contrast, the user may drag their thumb along their palm as if scrolling on a touchscreen device for vertical navigation. We refer to these as Swipe Left/Right (radial swipes) and Swipe Up/Down (palm swipes).
- **Pinch-and-Drag** gestures are performed by pinching the thumb and index finger together and dragging the hand upward or downward. Like swipe gestures, pinch-and-drag can be used to scroll through content or switch between screens. In fact, this form of interaction is commonly used in VR systems including the Meta Quest [28], HTC Vive [45], Microsoft HoloLens 2 [30], and Apple Vision Pro [2] for scrolling or repositioning.

We define gesture sets by pairing selection gestures with one or two axes of directional input (targeting gestures), enabling interaction with 1D and 2D interfaces, respectively. These gesture sets are summarized in Table 2.

3.2 Sensing Principle

Microgestures are intricate movements that primarily rely on the coordinated action of tendons in the hand, with minimal involvement of the wrist and forearm muscles. The flexor and

Table 2: The evaluated gesture sets for enabling 1D and 2D UI navigation.

	Finger Tap	Surface Tap	Swipe Left	Swipe Right	Swipe Up	Swipe Down	Pinch-and-Drag Up	Pinch-and-Drag Down
Swipe-1D-X	✓	✓	✓	✓	✗	✗	✗	✗
Swipe-1D-Y	✓	✓	✗	✗	✓	✓	✗	✗
PinchDrag-1D	✓	✓	✗	✗	✗	✗	✓	✓
Swipe-2D	✓	✓	✓	✓	✓	✓	✗	✗
Hybrid-2D	✓	✓	✓	✓	✗	✗	✓	✓

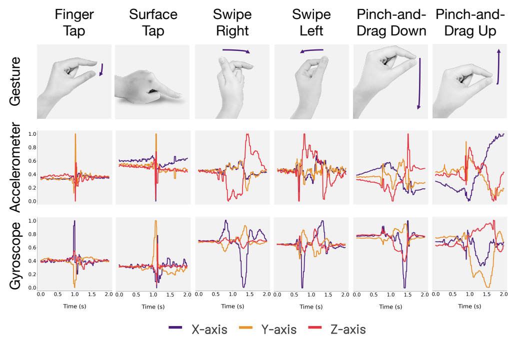


Figure 2: Illustrations of microgestures in the Hybrid-2D gesture set, along with corresponding normalized 6-axis IMU signals.

extensor tendons are responsible for bending and stretching the fingers, respectively, to facilitate precise finger motions. For example, radial swipes engage the extensor tendons along the back of the hand to guide the thumb's movement, while the flexor tendons in the palm facilitate fine control over the fingers. Although subtle, these microgestures generate mechanical forces in the tendons that slightly displace the soft tissue at the wrist.

Wrist-worn IMUs, such as those embedded in smartwatches, are capable of detecting these motions. As illustrated in Figure 2, microgestures produce distinct patterns in accelerometer and gyroscope data. However, the speed at which a gesture is performed directly influences the strength of these patterns, with slower movements generating lower-magnitude signals that are more difficult to classify reliably. This effect is particularly pronounced in swiping gestures, where the motion involves minimal displacement and relies heavily on rapid thumb movement for distinct inertial cues.

3.3 System Overview

Mudra employs the hierarchical approach illustrated in Figure 3. First, the system detects and categorizes microgestures into the broad categories listed in Section 3.1. It then disambiguates the direction of targeting gestures (Swipe, Pinch-and-Drag). To enable robust direction classification for swipes, Mudra only registers intentful swipes when they are performed at a sufficient speed; in other words, we use speed classification as a gating mechanism in the system design after a swipe is recognized.

Mudra uses a common deep learning backbone that is trained on publicly available IMU datasets to learn generalizable features about microgestures. This backbone is later fine-tuned for most of Mudra's intermediate stages using additional data that we collected in our research. The rest of this subsection describes the different steps in this pipeline.

3.3.1 Data Processing. The raw input for Mudra consists of 6-axis IMU data (3-axis accelerometer and 3-axis gyroscope). Although data was originally collected at 200 Hz, we empirically found that downsampling to 50 Hz maintained comparable performance while reducing computational overhead. After downsampling, the data is segmented into 2-second windows (stride = 250 ms), which are then processed through a high-pass filter with a cutoff frequency of 0.2 Hz to remove low-frequency noise and the gravity vector from the accelerometer data. Following this, a 3-sample median filter is applied to reduce transient noise and artifacts. Finally, Mudra separately normalizes the accelerometer and gyroscope data to ensure consistent scaling across all axes. This results in a processed input vector of dimensions 100×6.

3.3.2 Gesture Detection. Once the sensor data is processed, Mudra first determines whether a gesture is present in a given window, rejecting false activations from other general hand movements. The gesture detector is implemented as a lightweight single-layer convolutional neural network (CNN), optimized for low-latency operation. The model architecture consists of a single convolutional layer with 64 filters, followed by a fully connected layer with 128 units, and a final output layer with softmax activation to classify the presence or absence of a gesture.

3.3.3 Hierarchical Gesture Classification. Gesture classification in Mudra follows a hierarchical structure comprising four stages: (1) gesture category classification, (2) pinch-and-drag direction classification, (3) swipe speed classification, and (4) swipe direction classification. Each stage is implemented using a modified EfficientNet architecture [43], adapted for temporal modeling of 6-channel IMU data. While EfficientNet is originally designed for 2D image inputs, we introduce several modifications to better suit subtle motion signals:

(1) We replace all 2D spatial convolutions with 1D temporal convolutions, treating time as the primary dimension and preserving correlations across IMU channels.
(2) We simplify the original MBConv blocks into residual convolutional blocks consisting of two standard convolutional

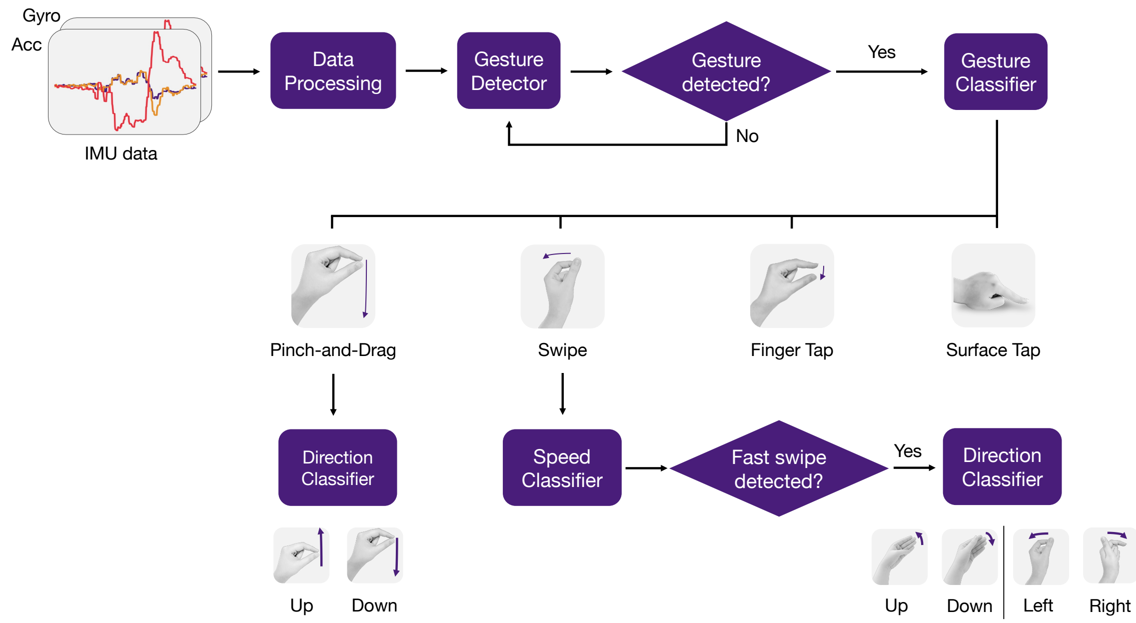


Figure 3: The hierarchical approach used by Mudra for microgesture recognition. The pipeline comprises five stages: (1) gesture detection, (2) gesture category classification, (3) pinch-and-drag direction classification, (4) swipe speed classification, and (5) swipe direction classification.

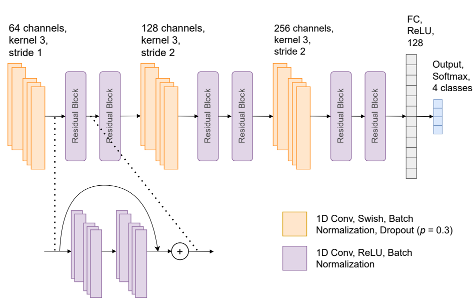


Figure 4: The modified EfficientNet [43] architecture used by Mudra for hierarchical gesture classification.

layers with batch normalization and ReLU activation. This modification reduces architectural complexity and improves stability when training on limited IMU data.

(3) While retaining the notion of compound scaling, we adapt it to the temporal setting by scaling the number of filters and network depth using parameters ($\alpha = 2.2$, $\beta = 2.1$, $\gamma = 1.15$), rather than jointly scaling spatial resolution.

Figure 4 illustrates the resulting architecture of this model. It begins with a 1D convolutional block, followed by a series of residual blocks, where each block consists of two convolutional layers with batch normalization and ReLU activation, allowing the model to capture intricate temporal features from the IMU data. The network progresses through three stages, each with increasing filter sizes (64, 128, and 256) and appropriate strides to downsample the feature maps, enhancing the model's ability to learn multi-scale patterns.

For feature aggregation, the model employs global average pooling to reduce the dimensionality of the output from the convolutional stages. This is followed by a fully connected layer with 128 units, incorporating ReLU activation to integrate learned features. The final output layer uses a softmax activation function to perform classification. The model's architecture ensures a balance between complexity and generalizability, with a dropout layer ($p = 0.3$) inserted after a subset of convolutional layers to prevent overfitting. We train the described network with an Adam optimizer (learning rate = 0.01) and cross-entropy loss.

3.3.4 Transfer Learning. To provide a solid backbone that can be applied to various stages in Mudra's hierarchical approach, we employ transfer learning using SparseIMU [40], a publicly available dataset designed for microgesture recognition using IMU sensors. Although SparseIMU contains data from a glove instrumented with multiple IMUs, we specifically use the sensor data from the "handback position", which closely resembles the placement of a smartwatch IMU. Furthermore, we select a subset of relevant gesture classes from SparseIMU that exhibit the most similar biomechanical characteristics to our gesture set: Static, Tap, Thumb Flexion / Extension, and Thumb Abduction / Adduction.

For pre-training, we adapt our proposed model architecture by modifying the softmax layer to match the number of selected SparseIMU classes. The model is then trained on the SparseIMU dataset. The encoder, consisting of all layers excluding the fully connected layers, serves as the pre-trained feature extractor. During downstream training on our microgesture dataset, we append task-specific fully connected layers to the pre-trained encoder. Rather than freezing the encoder layers, we adopt a fine-tuning strategy that allows the entire network to adapt to task-specific patterns. This approach facilitates the extraction of robust, transferable features.

3.3.5 Augmentation. To increase the dataset size and improve model generalizability, we apply three time-series augmentation methods tailored for gesture recognition [12]:

- **Rotation ($\theta \in [-45°, 45°]$)**, which simulates different hand orientations during gesture execution;
- **Scaling ($\lambda \in [0.8, 1.2]$)**, which simulates variations in gesture strength; and
- **Gaussian noise ($\sigma \in [0.01, 0.05]$)**, which replicates sensor noise and introduces randomness.

These augmentation techniques are applied randomly 10 times per sample with uniform sampling, creating natural variations in the data and enhancing the robustness of the model during training.

4 Study Design

We conducted a within-subjects study to collect wrist-worn IMU data as people performed a range of microgestures with controlled variations in direction and speed. The study protocol was approved by the Research Ethics Board at the University of Toronto (#42879).

4.1 Participants

Thirty participants (17 males, 13 females) aged between 20 and 38 years (mean age = 25.3 ± 4.2) were recruited for this study. We selected a diverse sample with varying wrist and hand sizes to ensure a fair evaluation of our wrist-worn system. The average wrist circumference among participants was 15.6 ± 1.4 cm, and the mean hand length was 18.3 ± 1.2 cm. The study took one hour to complete, and the participants were compensated $20 for their time and contribution.

4.2 Procedure

The study was conducted in a controlled and quiet room. Data was collected using a Google Pixel 2 smartwatch equipped with a custom WearOS app for collecting sensor data. The app recorded 6-axis IMU data at a sampling rate of 200 Hz. Participants wore the smartwatch on their right wrist as they completed four sessions of data collection. Although they were seated during the sessions, participants were free to change their posture and hand orientation at any moment (e.g., moving their arms above or below the desk, adjusting their seated position). After completing the second session, participants were instructed to remove and re-wear the smartwatch to simulate natural variations in device placement.

Participants completed three repetitions of every microgesture during each session. Before performing the repetitions for a single microgesture, participants were shown a video demonstrating the target action. After a 3-second countdown, three beeps were played at 2-second intervals to serve as timing cues and markers for segmenting the recorded data.

For targeting gestures, execution speed can influence how they are captured in IMU signals. While pinch-and-drag gestures produce stronger inertial signatures due to hand translation, swipes involve more subtle thumb motions and are therefore more sensitive to speed. To investigate the effect of speed on swipe recognition and direction classification, we asked participants to complete additional trials for all four swipes (left, right, up, and down) at different speeds in each of the four sessions. We controlled swipe speed by adjusting the gesture completion interval to 0.5 seconds for fast swipes and 1.5 seconds for slow swipes. The default 2-second interval remained as a timing cue between repetitions, but participants were instructed to use the shortened intervals for gesture completion.

To mitigate potential learning or order effects, we counterbalanced the presentation order of swipe directions, speeds, and other microgestures across participants and data collection sessions. Finally, to evaluate false positive rates and improve gesture detection robustness, we recorded an additional five-minute session of natural hand movements per participant, including activities such as typing, writing, trackpad manipulation, and free-form gestures.

4.3 Data Annotation

Following data collection, we reviewed the resulting dataset to ensure accurate segmentation and annotation. A custom data labeling tool was used to refine the initial timestamps and labels in light of delays between the presentation of gesture stimuli and participants' actual execution of microgestures. After excluding segments affected by imprecise performance or recording failures, the final dataset comprised 3,630 samples for analysis.

5 Results

In this section, we first present evidence supporting our proposed model architecture as Mudra's backbone. We then evaluate each component of Mudra's hierarchical pipeline to quantify task-specific performance, analyze sources of error, and motivate the use of swiping speed as a gating mechanism for direction classification. Finally, we combine these components to assess Mudra 's end-to-end classification performance across the five gesture sets introduced in Section 3.1 (**Swipe-1D-X**, **Swipe-1D-Y**, **PinchDrag-1D**, **Swipe-2D**, **Hybrid-2D**). All of our models were evaluated on segmented windows of IMU data and implemented in Python using `Keras`[1] with default values unless otherwise specified.

5.1 Model Selection

We compare our proposed model architecture against multiple deep learning models in the context of gesture category classification. Each model was trained to distinguish between Finger Tap, Surface Tap, Swipe, and Pinch-and-Drag; the latter two categories included all speed and direction variations as these attributes are disambiguated in subsequent stages of the pipeline. All models used the same 100×6 IMU data input, but with the following hyperparameters:

- **1D-CNN:** Number of convolutional layers = 3, number of filters for each layer = [64, 128, 256]
- **Gated Recurrent Units (GRU):** Number of GRU layers = 2, number of filters for each layer = [64, 128]
- **Bidirectional LSTM:** Number of Bi-LSTM layers = 2, number of filters for each layer = [64, 128].

[1] https://keras.io/

Table 3: The comparison of Mudra's modified EfficientNet backbone against other models in classifying four gesture categories: Finger Tap, Surface Tap, Swipe, and Pinch-and-Drag. All models were evaluated using 5-fold cross-validation. Performance is reported as means and standard deviations across participants.

Model	Accuracy	F1 Score
1D CNN	0.84 ± 0.02	0.82 ± 0.02
GRU	0.85 ± 0.01	0.83 ± 0.01
Bi-LSTM	0.79 ± 0.01	0.76 ± 0.02
CNN-LSTM	0.83 ± 0.00	0.81 ± 0.01
Inception-ResNet	0.84 ± 0.02	0.82 ± 0.02
Transformer	0.76 ± 0.02	0.72 ± 0.01
EfficientNet	0.86 ± 0.02	0.84 ± 0.03
Mudra	**0.87 ± 0.01**	**0.85 ± 0.01**
Mudra (w/ Transfer Learning)	**0.92 ± 0.05**	**0.87 ± 0.04**

- **CNN-LSTM:** Number of convolutional layers = 2, number of filters for each layer = [64, 128], number of LSTM layers = 1, number of filters for the LSTM layer = 64
- **Adapted Inception-ResNet [42]:** Number of inception modules = 4, number of filters for inception modules = 64
- **Transformer:** Number of layers = 2, number of heads = 6, feed-forward units = 256
- **Modified EfficientNet (Mudra) [43]**: $\phi = 1.0$, $\alpha = 1.2$, $\beta = 1.1$, $\gamma = 1.15$, number of convolutional layers = 3, number of filters for each layer = width × [32, 64, 128]

We trained all architectures from scratch using five-fold cross-validation and applied data augmentation to improve robustness and generalization.

Table 3 reports the accuracy and F1 scores of all the architectures. Our proposed model outperformed all baselines, achieving an accuracy of 0.87 ± 0.01 and an F1 score of 0.85 ± 0.01. Pre-training on the SparseIMU dataset [40] further increased accuracy and F1 score to 0.92 ± 0.05 and 0.87 ± 0.04, respectively. Thus, all subsequent experiments use this pre-trained model for evaluation.

5.2 Hierarchical Pipeline Evaluation

Next, we evaluated Mudra's performance across different components of its hierarchical pipeline. All models were trained using leave-one-subject-out cross-validation.

5.2.1 Gesture Detection and Category Classification. Mudra was able to differentiate gestures from periods of no movement or other non-gesture movements with a cross-user accuracy and F1 score of 0.95 ± 0.03 and 0.93 ± 0.04, respectively. For classifying the four gesture categories, the average accuracy and F1-score across all users were 0.93 ± 0.04 and 0.89 ± 0.06, respectively. The slight improvement in performance compared to the 5-fold cross-validation results can be attributed to the increased amount of training data, demonstrating the model's ability to perform reliably across new users.

Table 4: Mudra's ability to discriminate between different directions of the same targeting microgesture. All models were evaluated using leave-one-subject-out cross-validation. Performance is reported as means and standard deviations across participants.

Swipe Speed	Swipe Up / Down	Swipe Left / Right	All 4 Swipes
All	0.75 ± 0.08	0.75 ± 0.09	0.49 ± 0.08
Slow	0.71 ± 0.11	0.75 ± 0.10	0.45 ± 0.09
Fast	**0.88 ± 0.09**	**0.89 ± 0.10**	**0.61 ± 0.12**

5.2.2 Gesture Direction Classification. Next, we evaluated the performance of direction classification for both the targeting gestures: Pinch-and-Drag and Swipe. Although both gestures exhibit natural variation in execution speed, swipes are inherently more subtle and often generate weaker inertial signals, particularly when performed slowly, leading to increased ambiguity in direction classification. To investigate this effect, we stratified swipe direction classifiers by speed.

Mudra achieved an average accuracy of 0.96 ± 0.07 in distinguishing Pinch-and-Drag Up / Down, while the results for swipe direction classification, stratified by speed, are summarized in Table 4. Unsurprisingly, distinguishing between four swipe directions proved more challenging than binary classification. The confusion matrix for fast swipes shown in Figure 5 illustrates that the most commonly confused class pairs are Swipe Left / Down and Swipe Right / Up. This can be attributed to the similarity in wrist motion produced by these pairs of gestures. Swipe Right and Swipe Up, performed against the palm, start with an extended thumb that bends, resulting in similar flexor tendon motion. In contrast, Swipe Left and Swipe Down produce similar extensor tendon motions as the thumb straightens toward the end of the gesture. Binary swipe models performed comparably for swipes executed along the index finger and across the palm.

Regardless of the number of directions included in the model, swiping speed had a significant impact on performance. Incorporating slower swipes decreased the accuracy of the two-class models by approximately 15% and the accuracy of the four-class model by approximately 20%. When we assumed that slow swipes had been filtered out, Mudra was able to discriminate between Swipe Left and Right with an accuracy of 0.89 ± 0.10, Swipe Up and Down with an accuracy of 0.88 ± 0.09, and all four directions with an accuracy of 0.61 ± 0.12.

5.2.3 Swipe Speed Classification. The results from the previous experiment support the inclusion of a gating mechanism that restricts swipe direction classification to swipes with sufficient speed. To enable this, we trained a binary classifier to distinguish between fast and slow swipes, achieving accuracies of 0.88 ± 0.10 for Swipe Left / Right, 0.90 ± 0.08 for Swipe Up / Down, and 0.86 ± 0.08 when evaluated across all four directions.

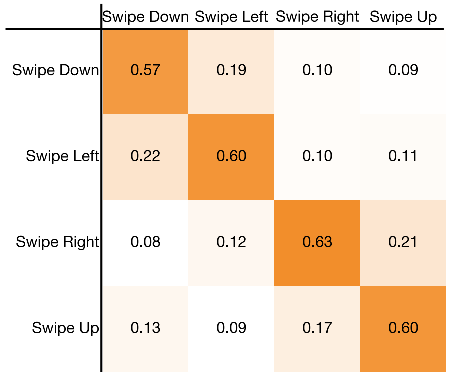


Figure 5: Confusion matrix showing Mudra's performance in classifying all four swipe directions. The model was evaluated using leave-one-subject-out cross-validation.

Table 5: Mudra's ability to support different gesture sets. All models were evaluated using leave-one-subject-out cross-validation. Performance is reported as means and standard deviations across participants.

Gesture Set	Accuracy	F1 Score	Precision	Recall	False Positive Rate
Swipe-1D-X	0.86 ± 0.07	0.86 ± 0.07	0.87 ± 0.07	0.86 ± 0.07	0.03 ± 0.03
Swipe-1D-Y	0.85 ± 0.07	0.85 ± 0.08	0.87 ± 0.08	0.85 ± 0.08	0.02 ± 0.02
PinchDrag-1D	**0.93 ± 0.06**	**0.93 ± 0.06**	**0.95 ± 0.05**	**0.94 ± 0.06**	**0.02 ± 0.02**
Swipe-2D	0.64 ± 0.09	0.63 ± 0.10	0.65 ± 0.12	0.64 ± 0.10	0.04 ± 0.03
Hybrid-2D	**0.86 ± 0.07**	**0.85 ± 0.07**	**0.88 ± 0.07**	**0.86 ± 0.07**	**0.04 ± 0.03**

5.3 Overall System Performance

So far, we have evaluated the performance of individual components of Mudra. In the complete system, however, the accuracy of one stage depends on the performance of the previous ones. Therefore, we conclude our evaluation by assessing Mudra's end-to-end classification performance across the different gesture sets using leave-one-subject-out cross-validation. Similar to prior work on microgesture sensing [3, 7, 18, 50], we estimate false positive rates using the separate session of non-gesture activity data we collected from each participant.

Table 5 summarizes the end-to-end classification performance across the different gesture sets supported by Mudra, while the confusion matrices for the best-performing 1D and 2D gesture sets are illustrated in Figure 6. Among the three 1D navigation gesture sets, **Swipe-1D-X** and **Swipe-1D-Y** performed comparably, yielding F1 scores of 0.86 ± 0.07 and 0.85 ± 0.08, respectively. **PinchDrag-1D** achieved a higher F1 score of 0.93 ± 0.06, likely due to the more pronounced and consistent inertial signatures produced by hand translation during the pinch-and-drag gesture. The two 2D navigation sets, **Swipe-2D** and **Hybrid-2D**, showed more divergent results. **Swipe-2D** achieved an F1 score of 0.63 ± 0.10, with the primary challenge arising from the difficulty of distinguishing all four swipe directions within a single classification task. **Hybrid-2D**, on the other hand, simplified the classification problem by relying on two separate microgesture categories for different directional inputs, resulting in a substantially improved F1 score of 0.85 ± 0.07. The observed false positive rates were low across all gesture sets (<0.05) and were comparable to prior studies, including 0.02 for VibAware [18] and 0.04 for AO-Finger [50].

6 Discussion

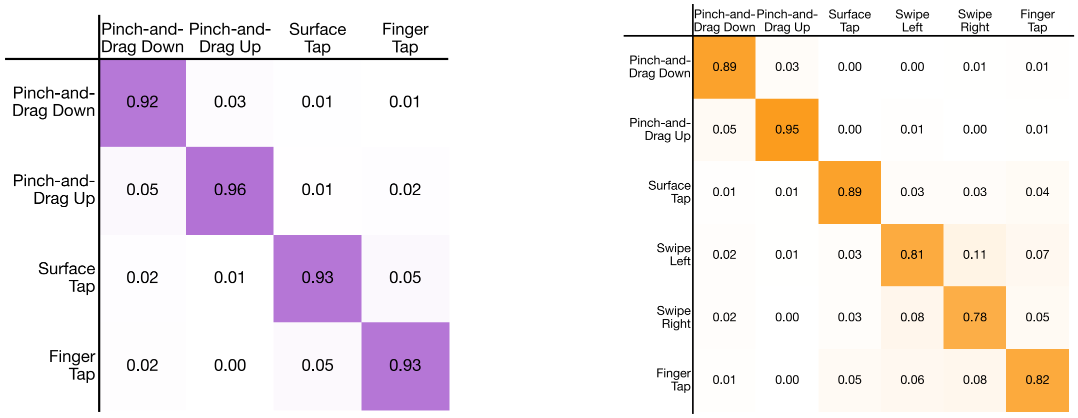
In this section, we first summarize the key findings of this work and its implications for application design. We then elaborate on the applications and scenarios that would most benefit from a system like Mudra. Finally, we identify some limitations of our work that currently limit its practical adoption, highlighting opportunities for future exploration.

6.1 Key Findings

Our work focused on fundamental microgestures, such as Finger Tap and Swipe, that can facilitate a wide range of interactions akin to touchscreen input. We also considered additional microgestures, Surface Tap and Pinch-and-Drag, that complement the functionality of these gestures. Building on these four gestures, we evaluated five candidate gesture sets designed for 1D and 2D UI navigation, uncovering important design trade-offs between gesture complexity, directional coverage, and classification accuracy. Beyond achieving high performance in microgesture classification, our analysis highlights Mudra's ability to accurately characterize modifiers for swiping gestures, including their direction and speed. These modifiers represent an important step towards enabling seamless, continuous interactions.

For 1D navigation, the three evaluated sets — **Swipe-1D-X**, **Swipe-1D-Y** and **PinchDrag-1D** — performed reliably, with **PinchDrag-1D** achieving the highest performance due to the stronger inertial signatures generated by hand translation. For 2D navigation, however, we observed a clear distinction between **Swipe-2D** and **Hybrid-2D**. While **Swipe-2D** supported four-way directional control using only swipe gestures, it resulted in lower overall performance due to confusion between directions. In contrast, **Hybrid-2D** achieved substantially better accuracy by leveraging two distinct microgesture categories — swipes and pinch-and-drags — along orthogonal axes, suggesting that incorporating multiple gesture categories can improve reliability for multidimensional interfaces. Further, the leave-one-subject-out results for all the experiments suggest that Mudra can generalize across users without requiring explicit per-user calibration, making it suitable for out-of-the-box use.

Beyond classification accuracy, we explored direction and speed as two modifiers for swiping gestures. Consistent with prior work, we found that direction classification degrades for slower swipes,



(a) PinchDrag-1D **(b) Hybrid-2D**

Figure 6: Confusion matrices for the best-performing 1D and 2D UI navigation gesture sets. Both models were evaluated using leave-one-subject-out cross-validation.

motivating our design choice to prioritize swipes performed at moderate speeds for reliable directional input. While Mudra does not explicitly use speed as a control parameter, our findings suggest that swipe speed could serve as a useful signal for modulating interaction in future systems, potentially supporting rate-based control where faster swipes correspond to faster navigation. Across all experiments, the leave-one-subject-out results further indicate that Mudra generalizes well to unseen users without requiring explicit person-specific calibration.

6.2 Potential Applications

Although gestures can be used in all sorts of environments, microgestures are particularly well-suited for scenarios that require sustained or discreet use. We describe a few of these applications below.

6.2.1 Extended Reality Control. Physical controllers have largely been acceptable for virtual reality (VR) systems because they are used in dedicated spaces with minimal interruption. With augmented reality (AR) and smartglasses, controllers go against the goal of having systems seamlessly integrate with the user's surroundings because they occupy the user's hands. Innovations in visual hand tracking have made it possible for users to control interfaces with natural mid-air gestures like pointing and grabbing [19, 41], but to use these techniques in any XR system, users must hold their hands in front of an outward-facing camera from the headset. This can lead to a form of fatigue known as "gorilla arm" [10] that precludes long-term interactions as well as requires headsets instrumented with cameras to capture hand motion. Microgestures sensed by a smartwatch can alleviate this problem by allowing people to keep their hands to their sides as they make subtle, socially acceptable gestures to control mobile AR. The interfaces for AR applications also tend to be simple [16], requiring selection and targeting primitives to navigate. From Mudra's gesture set, Finger Tap and Surface Tap interactions could be used for selection, while Pinch-and-Drag or Swipe interactions could enable scrolling.

6.2.2 Dual-Device Interactions. Within the human-computer interaction community, researchers have proposed various taxonomies for dual-device interactions. For example, Chen et al. [7] proposed various ways that a smartwatch and a smartphone could serve different roles in a shared application. One of the paradigms they proposed was having the smartphone serve as the primary interface while having the smartwatch be used as a tool palette. By relegating the tool palette to a separate device, users would not need to sacrifice screen space or remove themselves from the primary interface altogether to switch functions. While Chen et al.'s work sought to leverage both the visual and tactile affordances of smartphones and smartwatches, microgestures detected from a smartwatch using Mudra could still be used to take over secondary functions from a smartphone. A follow-up study by Kubo et al. [20] proposed that dual-device interactions could even be made context-aware by sensing users' grip and arm posture. For instance, they proposed different allocations of functionality depending on whether the user's smartwatch is at their side or raised in front of their body, and Mudra could be used to a similar effect.

6.2.3 Encumbrance. Beyond dual-device interactions, there are many situations when users only have one hand free to interact with a single device. Carrying groceries or a handbag while interacting with a smartphone creates a temporary situational impairment [33, 37] when people must use the same hand to both secure and touch the screen. In these situations, Mudra could help by precluding the need for holding the device itself, rendering similar benefits as touch-sensitive wireless earbuds but without the need to raise one's hand to their ear. Sharma et al. [38, 39] explored different scenarios of encumbrance when individuals are interested in performing gestures while holding everyday objects ranging from cups and plates to pencils and papers. Although it is difficult to perform large mid-air gestures in these scenarios, they identified many opportunities for small microgestures that only required the thumb, index, or middle finger. Mudra could support many of these scenarios provided that the user is still able to grasp the object.

6.3 Limitations

The unique advantages of the gestures recognized by Mudra suggest its potential for large-scale use, yet several factors currently limit its scalability. First, Mudra relies on IMU signals, so motions from daily activities like walking and exercise generate artifacts that can distort the signal. A structured dataset collected from mobile scenarios, together with modeling that either accounts for confounding movements or conditions inference upon them, could mitigate this issue. Incorporating additional modalities like PPG could also enhance robustness in dynamic environments. Many commercial smartwatches (e.g., Apple Watch, Samsung Galaxy Watch) use PPG to detect a limited set of macrogestures, but we found it to be less effective for microgesture detection during our initial testing, possibly due to its low sampling rate. Future work could investigate high-resolution PPG signals to facilitate multimodal microgesture detection.

Another factor related to Mudra's scalability is its ability to operate in various environments and scenarios. One unique advantage of microgestures is their ability to facilitate interactions in scenarios where users are otherwise compromised. This includes situations such as when users are holding objects, mentally occupied with tasks, or performing activities like washing utensils. To fully leverage this advantage and enhance the robustness of Mudra, a more extensive evaluation across diverse contexts is required. Recent advancements in model personalization [51] present promising alternatives to traditional methods by enabling effective personalization with just a few examples, thus reducing the need for extensive calibration and potentially enhancing model performance. We believe that integrating such techniques could further improve Mudra's generalizability and performance, leading to better microgesture classification across diverse users and contexts.

7 Conclusion

In conclusion, our work demonstrates that Mudra can accurately detect and classify microgestures with variation in their speed and direction. These affordances, typically achieved using custom hardware designed for fine-grained motion detection, were successfully validated using motion sensors available on commodity smartwatches. By leveraging this existing platform, Mudra provides a step toward enabling the millions of users who already own a smartwatch device to have microgesture control right at their fingertips.

References

[1] Sonu Agarwal, Sanjay Ghosh, and Gaurav Gupta. 2017. Evaluation of microgesture recognition using a smartwatch. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 986–991.

[2] Apple. 2025. Design for spatial input. https://developer.apple.com/videos/play/wwdc2023/10073. [Accessed 29-03-2025].

[3] Adrien Chaffangeon Caillet, Alix Goguey, and Laurence Nigay. 2023. μGlyph: a Microgesture Notation. 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 269, 28 pages. doi:10.1145/3544548.3580693

[4] Edwin Chan, Teddy Seyed, Wolfgang Stuerzlinger, Xing-Dong Yang, and Frank Maurer. 2016. User elicitation on single-hand microgestures. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 3403–3414.

[5] 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. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 255–260.

[6] Taizhou Chen, Tianpei Li, Xingyu Yang, and Kening Zhu. 2023. Efring: Enabling thumb-to-index-finger microgesture interaction through electric field sensing using single smart ring. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 4 (2023), 1–31.

[7] Xiang 'Anthony' Chen, Tovi Grossman, Daniel J Wigdor, and George Fitzmaurice. 2014. Duet: exploring joint interactions on a smart phone and a smart watch. In *Proceedings of the SIGCHI Conference on human factors in computing systems*. 159–168.

[8] Masaaki Fukumoto and Yoshinobu Tonomura. 1997. "Body coupled FingerRing" wireless wearable keyboard. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. 147–154.

[9] Sarthak Ghosh, Hyeong Cheol Kim, Yang Cao, Arne Wessels, Simon T Perrault, and Shengdong Zhao. 2016. Ringteraction: coordinated thumb-index interaction using a ring. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 2640–2647.

[10] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1063–1072.

[11] Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. 2016. DigitSpace: Designing Thumb-to-Fingers Touch Interfaces for One-Handed and Eyes-Free Interactions. 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, 1526–1537. doi:10.1145/2858036.2858483

[12] Guillermo Iglesias, Edgar Talavera, Ángel González-Prieto, Alberto Mozo, and Sandra Gómez-Canaval. 2023. Data augmentation techniques in time series domain: a survey and taxonomy. *Neural Computing and Applications* 35, 14 (2023), 10123–10145.

[13] Yasha Iravantchi, Mayank Goel, and Chris Harrison. 2019. BeamBand: Hand Gesture Sensing with Ultrasonic Beamforming. 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, 1–10. doi:10.1145/3290605.3300245

[14] Nikhita Joshi, Parastoo Abtahi, Raj Sodhi, Nitzan Bartov, Jackson Rushing, Christopher Collins, Daniel Vogel, and Michael Glueck. 2023. Transferable Microgestures Across Hand Posture and Location Constraints: Leveraging the Middle, Ring, and Pinky Fingers. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–17.

[15] Chirag Kandoi, Changsoo Jung, Sheikh Mannan, Hannah VanderHoeven, Quincy Meisman, Nikhil Krishnaswamy, and Nathaniel Blanchard. 2023. Intentional microgesture recognition for extended human-computer interaction. In *International Conference on Human-Computer Interaction*. Springer, 499–518.

[16] Chirag Kandoi, Changsoo Jung, Sheikh Mannan, Hannah VanderHoeven, Quincy Meisman, Nikhil Krishnaswamy, and Nathaniel Blanchard. 2023. Intentional Microgesture Recognition for Extended Human-Computer Interaction. In *Human-Computer Interaction: Thematic Area, HCI 2023, Held as Part of the 25th HCI International Conference, HCII 2023, Copenhagen, Denmark, July 23–28, 2023, Proceedings, Part I* (Copenhagen, Denmark). Springer-Verlag, Berlin, Heidelberg, 499–518. doi:10.1007/978-3-031-35596-7_32

[17] Hsin-Liu Kao, Artem Dementyev, Joseph A Paradiso, and Chris Schmandt. 2015. NailO: fingernails as an input surface. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 3015–3018.

[18] Jina Kim, Minyung Kim, Woo Suk Lee, and Sang Ho Yoon. 2023. VibAware: Context-Aware Tap and Swipe Gestures Using Bio-Acoustic Sensing. In *Proceedings of the 2023 ACM Symposium on Spatial User Interaction* (Sydney, NSW, Australia) *(SUI '23)*. Association for Computing Machinery, New York, NY, USA, Article 6, 12 pages. doi:10.1145/3607822.3614544

[19] Kenrick Kin, Chengde Wan, Ken Koh, Andrei Marin, Necati Cihan Camgöz, Yubo Zhang, Yujun Cai, Fedor Kovalev, Moshe Ben-Zacharia, Shannon Hoople, Marcos Nunes-Ueno, Mariel Sanchez-Rodriguez, Ayush Bhargava, Robert Wang, Eric Sauser, and Shugao Ma. 2024. STMG: A Machine Learning Microgesture Recognition System for Supporting Thumb-Based VR/AR Input. 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 753, 15 pages. doi:10.1145/3613904.3642702

[20] Yuki Kubo, Ryosuke Takada, Buntarou Shizuki, and Shin Takahashi. 2017. Exploring context-aware user interfaces for smartphone-smartwatch cross-device interaction. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–21.

[21] Falko Kuester, Michelle Chen, Mark E Phair, and Carsten Mehring. 2005. Towards keyboard independent touch typing in VR. In *Proceedings of the ACM symposium on Virtual reality software and technology*. 86–95.

[22] 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

[23] Chi-Jung Lee, Ruidong Zhang, Devansh Agarwal, Tianhong Catherine Yu, Vipin Gunda, Oliver Lopez, James Kim, Sicheng Yin, Boao Dong, Ke Li, Mose Sakashita, Francois Guimbretiere, and Cheng Zhang. 2024. EchoWrist: Continuous Hand Pose Tracking and Hand-Object Interaction Recognition Using Low-Power Active Acoustic Sensing On a Wristband. 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 403, 21 pages. doi:10.1145/3613904.3642910

[24] Guangchuan Li, David Rempel, Yue Liu, Weitao Song, and Carisa Harris Adamson. 2021. Design of 3D Microgestures for Commands in Virtual Reality or Augmented Reality. *Applied Sciences* 11, 14 (2021). doi:10.3390/app11146375

[25] Jeffrey Lin, Carisa Harris-Adamson, and David Rempel. 2019. The design of hand gestures for selecting virtual objects. *International Journal of Human–Computer Interaction* 35, 18 (2019), 1729–1735.

[26] Fujun Ma, Fanghao Song, Yan Liu, and Jiahui Niu. 2020. Quantitative analysis on the interaction fatigue of natural gestures. *IEEE Access* 8 (2020), 190797–190811.

[27] Jess McIntosh, Charlie McNeill, Mike Fraser, Frederic Kerber, Markus Löchtefeld, and Antonio Krüger. 2016. EMPress: Practical hand gesture classification with wrist-mounted EMG and pressure sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2332–2342.

[28] Meta. 2025. Learn about Hand and Body Tracking on Meta Quest | Quest Help. https://www.meta.com/help/quest/290147772643252/. [Accessed 28-03-2025].

[29] Meta. 2025. Orion AI Glasses: The Future of AR Glasses Technology. https://about.meta.com/realitylabs/orion/. [Accessed 29-03-2025].

[30] Microsoft. 2025. Point and commit with hands - Mixed Reality. https://learn.microsoft.com/en-us/windows/mixed-reality/design/point-and-commit. [Accessed 29-03-2025].

[31] Sam Miller, Andy Smith, Sina Bahram, and Robert St. Amant. 2012. A glove for tapping and discrete 1D/2D input. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*. 101–104.

[32] Weizhi Nai, David Rempel, Yue Liu, Alan Barr, Carisa Harris-Adamson, and Yongtian Wang. 2017. Performance and user preference of various functions for mapping hand position to movement velocity in a virtual environment. In *Virtual, Augmented and Mixed Reality: 9th International Conference, VAMR 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9-14, 2017, Proceedings 9*. Springer, 141–152.

[33] Alexander Ng, Stephen A Brewster, and John H Williamson. 2014. Investigating the effects of encumbrance on one-and two-handed interactions with mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1981–1990.

[34] Farshid Salemi Parizi, Eric Whitmire, and Shwetak Patel. 2019. Auraring: Precise electromagnetic finger tracking. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 4 (2019), 1–28.

[35] Anna Peshock, Julia Duvall, and Lucy E Dunne. 2014. Argot: A wearable one-handed keyboard glove. In *Proceedings of the 2014 ACM international symposium on wearable computers: adjunct program*. 87–92.

[36] Manuel Prätorius, Dimitar Valkov, Ulrich Burgbacher, and Klaus Hinrichs. 2014. DigiTap: an eyes-free VR/AR symbolic input device. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology* (Edinburgh, Scotland) *(VRST '14)*. Association for Computing Machinery, New York, NY, USA, 9–18. doi:10.1145/2671015.2671029

[37] Zhanna Sarsenbayeva, Niels van Berkel, Chu Luo, Vassilis Kostakos, and Jorge Goncalves. 2017. Challenges of situational impairments during interaction with mobile devices. In *Proceedings of the 29th australian conference on computer-human interaction*. 477–481.

[38] Adwait Sharma, Michael A Hedderich, Divyanshu Bhardwaj, Bruno Fruchard, Jess McIntosh, Aditya Shekhar Nittala, Dietrich Klakow, Daniel Ashbrook, and Jürgen Steimle. 2021. SoloFinger: Robust microgestures while grasping everyday objects. In *Proceedings of the 2021 CHI conference on human factors in computing systems*. 1–15.

[39] Adwait Sharma, Joan Sol Roo, and Jürgen Steimle. 2019. Grasping microgestures: Eliciting single-hand microgestures for handheld objects. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.

[40] Adwait Sharma, Christina Salchow-Hömmen, Vimal Suresh Mollyn, Aditya Shekhar Nittala, Michael A. Hedderich, Marion Koelle, Thomas Seel, and Jürgen Steimle. 2023. SparseIMU: Computational Design of Sparse IMU Layouts for Sensing Fine-grained Finger Microgestures. *ACM Trans. Comput.-Hum. Interact.* 30, 3, Article 39 (June 2023), 40 pages. doi:10.1145/3569894

[41] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, et al. 2015. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 3633–3642.

[42] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. 2016. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. arXiv:1602.07261 [cs.CV] https://arxiv.org/abs/1602.07261

[43] Mingxing Tan. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946* (2019).

[44] Shari Trewin. 2008. Physical impairment. *Web accessibility: a foundation for research* (2008), 37–46.

[45] HTC Vive. 2025. Hand tracking. https://www.vive.com/us/support/focus3/category_howto/hand-tracking.html. [Accessed 29-03-2025].

[46] Anandghan Waghmare, Roger Boldu, Eric Whitmire, and Wolf Kienzle. 2023. OptiRing: Low-Resolution Optical Sensing for Subtle Thumb-to-Index Micro-Interactions. In *Proceedings of the 2023 ACM Symposium on Spatial User Interaction* (Sydney, NSW, Australia) *(SUI '23)*. Association for Computing Machinery, New York, NY, USA, Article 8, 13 pages. doi:10.1145/3607822.3614538

[47] 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*. 3847–3851.

[48] Eric Whitmire, Mohit Jain, Divye Jain, Greg Nelson, Ravi Karkar, Shwetak Patel, and Mayank Goel. 2017. Digitouch: Reconfigurable thumb-to-finger input and text entry on head-mounted displays. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–21.

[49] Katrin Wolf and Jonas Willaredt. 2015. PickRing: seamless interaction through pick-up detection. In *Proceedings of the 6th augmented human international conference*. 13–20.

[50] Chenhan Xu, Bing Zhou, Gurunandan Krishnan, and Shree Nayar. 2023. AO-Finger: Hands-free Fine-grained Finger Gesture Recognition via Acoustic-Optic Sensor Fusing. 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 306, 14 pages. doi:10.1145/3544548.3581264

[51] Xuhai Xu, Jun Gong, Carolina Brum, Lilian Liang, Bongsoo Suh, Shivam Kumar Gupta, Yash Agarwal, Laurence Lindsey, Runchang Kang, Behrooz Shahsavari, Tu Nguyen, Heriberto Nieto, Scott E Hudson, Charlie Maalouf, Jax Seyed Mousavi, and Gierad Laput. 2022. Enabling Hand Gesture Customization on Wrist-Worn Devices. 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 496, 19 pages. doi:10.1145/3491102.3501904

[52] Xing-Dong Yang, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. 2012. Magic finger: always-available input through finger instrumentation. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 147–156.

[53] Sang Ho Yoon, Ke Huo, Vinh P Nguyen, and Karthik Ramani. 2015. TIMMi: Finger-worn textile input device with multimodal sensing in mobile interaction. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*. 269–272.

[54] Sang Ho Yoon, Ke Huo, and Karthik Ramani. 2014. Plex: finger-worn textile sensor for mobile interaction during activities. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. 191–194.

[55] Boning Zhang, Yiqiang Chen, Yueliang Qian, and Xiangdong Wang. 2011. A ring-shaped interactive device for large remote display and mobile device control. In *Proceedings of the 13th international conference on Ubiquitous computing*. 473–474.

[56] Cheng Zhang, Anandghan Waghmare, Pranav Kundra, Yiming Pu, Scott Gilliland, Thomas Ploetz, Thad E Starner, Omer T Inan, and Gregory D Abowd. 2017. FingerSound: Recognizing unistroke thumb gestures using a ring. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–19.

[57] Yang Zhang and Chris Harrison. 2015. Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 167–173.