RadarNet: Efficient Gesture Recognition Technique Utilizing a Miniature Radar Sensor

Eiji Hayashi
eijihayashi@google.com
Google
Mountain View, California

Jaime Lien
jaimelien@google.com
Google
Mountain View, California

Nicholas Gillian
ngillian@google.com
Google
Mountain View, California

Leonardo Giusti
lgiusti@google.com
Google
Mountain View, California

Dave Weber
daveweber@google.com
Google
Mountain View, California

Jin Yamanaka
yamanakaj@google.com
Google
Mountain View, California

Lauren Bedal
lbedal@google.com
Google
Mountain View, California

Ivan Poupyrev
ipoupyrev@google.com
Google
Mountain View, California

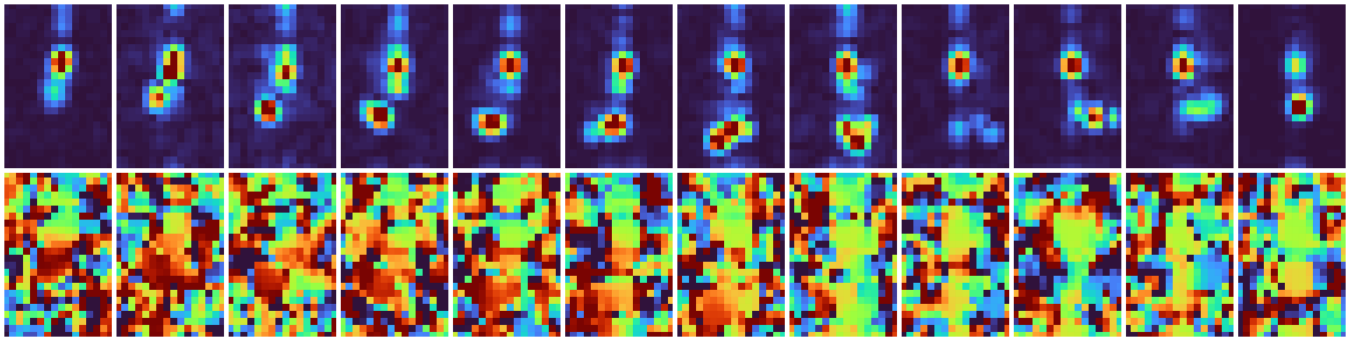


Figure 1: A time series visualization of a left swipe motion with absolute range Doppler maps (top) and interferometry maps (bottom). A hand moving close to a Soli chip round 6th frame, then, moving away. In the interferometry maps, the color of cells s at the same position as hand above changed from red to green, and to blue, showing the hand moving right to left.

ABSTRACT

Gestures are a promising candidate as an input modality for ambient computing where conventional input modalities such as touchscreens are not available. Existing works have focused on gesture recognition using image sensors. However, their cost, high battery consumption, and privacy concerns made cameras challenging as an always-on solution. This paper introduces an efficient gesture recognition technique using a miniaturized 60 GHz radar sensor. The technique recognizes four directional swipes and an omni-swipe using a radar chip (6.5 × 5.0 mm) integrated into a mobile phone. We developed a convolutional neural network model efficient enough for battery powered and computationally constrained processors. Its model size and inference time is less than 1/5000 compared to an existing gesture recognition technique using radar. Our evaluations with large scale datasets consisting of 558,000 gesture samples and 3,920,000 negative samples demonstrated our algorithm's efficiency, robustness, and readiness to be deployed outside of research laboratories.

CCS CONCEPTS

• **Human-centered computing** → **Gestural input**; *Smartphones.*

KEYWORDS

gesture recognition; mobile; deep learning; radar sensing

ACM Reference Format:
Eiji Hayashi, Jaime Lien, Nicholas Gillian, Leonardo Giusti, Dave Weber, Jin Yamanaka, Lauren Bedal, and Ivan Poupyrev. 2021. RadarNet: Efficient Gesture Recognition Technique Utilizing a Miniature Radar Sensor. In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3411764.3445367

CHI '21, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8096-6/21/05...$15.00
https://doi.org/10.1145/3411764.3445367

1 INTRODUCTION

A new generation of consumer products, such as voice-activated speakers [1, 12], intelligent thermostats [13], and interactive garments [14], has brought some of the promises of ambient computing systems into our homes, helping in the background while we focus on our daily tasks and routines. As Weiser pointed out in his seminal paper on calm technology [36], "if computers are going to be everywhere, they better stay out of the way." In a world where technology is becoming omnipresent, designing interactions with products that live at the periphery of attention is highly relevant.

Gestural input is a promising candidate for ambient computing interactions [34]: it opens the opportunity to design interactions that are eye-free and less cognitively demanding, allowing users to access information and services while staying engaged in their primary tasks (e.g. cooking, driving) [18, 28]. However, from early research work at MIT [3] to more recent product launches such as Microsoft Kinect [24], most of the work on gesture recognition techniques has focused on screen interactions, which often require full user attention as primary tasks. When considering gestural interaction in ambient computing contexts, we need to consider a different set of technological and interaction requirements, since users will interact with devices at the periphery of their attention:

- **Always-on**: gesture recognition techniques should run continuously, and be ready anytime the user wants to initiate the interaction. The value of gesture-based interactions in ambient computing systems is in their immediacy: the user can quickly interact with peripheral devices for simple tasks, with minimal cognitive effort, and without complex hand-eye coordination. Any friction, such as the need to wake-up the device, will impact the usefulness of gestural interactions in these contexts.
- **Reliable**: gesture recognition techniques for ambient computing applications should work in a variety of different contexts. These devices can be worn (e.g., smart watch), carried (e.g., mobile phone), or fixed within a given environment (e.g., display on a nightstand). They should be robust against environmental changes such as air temperature and lighting conditions.
- **Private**: the diffusion of products with advanced sensing capabilities in personal spaces, such as our bedrooms, living rooms or workplaces, makes privacy a key factor for their wide adoption.
- **Small**: gesture recognition techniques should have a small footprint in order to be embedded in a variety of objects without compromising their form factor or aesthetic.
- **Invisible**: such techniques should disappear behind surfaces, without requiring openings or other modifications to the physical design of the product. This new generation of devices is designed to share our home environments, and the quality of industrial design is a key factor for their adoption.

This paper introduces RadarNet, a gesture recognition technique using a radar sensing technology, Soli, that has multiple attractive properties for the deployment of gesture interactions in ambient computing. Soli is based on a custom-designed solid state BiCMOS radar sensor chip[1] which is inexpensive and small enough to integrate into space-constrained devices. The radar RF signal propagates through plastics, glass, and other non-metallic materials; thus, we can invisibly place the chip inside device enclosures. The RF wave is not affected by ambient light or noise. Soli is less privacy invasive compared to image-based sensors since the radar does not produce distinguishable representations of a target's spatial structure. Finally, Soli's sensitivity to sub-millimeter displacements regardless of distance allows motion recognition in both near and far fields with the same hardware.

Leveraging these unique advantages of radar-based sensing, we developed an interactive technique that recognizes four directional gestures (right, left, up, and down) and omni-swipes, i.e., swipe motions in any direction, including diagonal.

In summary, the contributions of this paper are as follows:

- The development of a radar-based semiconductor sensor with an extraordinarily small footprint that allows integration of the chip into devices with tight form factor constraints (e.g., mobile phones, smart watches).
- The development of RadarNet, a novel algorithm that recognizes gestures with unsegmented time series radar signals with tiny computational resource consumption. Our model's size and inference time is less than 0.02% of the existing work [35], allowing the algorithm to run on battery powered devices.
- Presenting the design rationale for the deep neural network structure used in the proposed algorithm, which can be generalized to development of other radar-based gesture recognition systems.
- The evaluation of the proposed algorithm with a large scale dataset that is a few hundred times bigger than the datasets used in existing work on radar-based gesture recognition techniques. Leveraging the large scale dataset, we conducted novel evaluations such as performance analysis on a task where an algorithm detects gestures from unsegmented data streams and the impact of dataset sizes on gesture recognition performance.
- Finally, we present a set of use cases for our gesture recognition technique that outline new opportunities for this input method in the context of ambient computing.

2 RELATED WORK

Conventional input methods such as mouse, keyboard, and touchscreen are the most commonly used to execute complex tasks with computational devices. In the context of ambient computing applications, alternative input methods, such as voice and gesture interactions, are becoming increasingly popular. These modalities allow the user to access and manipulate digital information while staying engaged with other, often more important, tasks (e.g., cooking and dining).

Voice interfaces are the most widely adopted alternative due to the proliferation of personal assistants on mobile phones (e.g., Google Assistant [11] and Siri [2]) and smart speakers (e.g., Amazon

[1]The Soli BiCMOS radar sensor was co-designed in collaboration with Infineon Corporation [27, 33]

Echo [1] and Nest Home Hub [12]). Although these voice-based interactions are promising, the commands are often lengthy; users have to pronounce hot-words before initiating the interaction; and in some contexts such as in quiet places and during conversations, social acceptability might be an issue for certain users.

Gestural interactions have been studied in the HCI community, often using camera-based sensing systems [26]. Early work on 3D hand pose estimation used depth images (e.g., [22, 30]). More recently, with advancements in machine learning, a large amount of work has focused on 3D pose estimation of the hand, including occluded parts, using standard RGB cameras [9, 25]. Other recent work has enabled real time hand pose estimation on mobile phones with their built-in cameras [38]. However, because of their limited field of view, cameras cannot detect hands far from the sensor's line of sight. Moreover, the power consumption of the camera prevents running for long durations on battery powered devices. Finally, there are strong privacy concerns around keeping cameras always on (e.g., [5, 37]). These limitations make it challenging to use cameras as an input method for ambient computing applications, which require always-on sensing systems that preserve user privacy.

To address these challenges, a number of works have explored other sensing modalities, including IR proximity sensors [4] and built-in magnetometers on mobile phones detecting magnetic fields generated by magnets attached to a fingertip [16] or controllers [18]. Humantenna recognized twelve whole body gestures based on a voltage measured at one place on a user's body, using the whole body as an antenna receiving surrounding electric noise [8]. Soundwave used a built-in speaker and microphone on a laptop to detect gestures with ultrasound [15].

Other research has leveraged existing wireless signals (e.g., [6]). WiSee recognized nine whole body gestures at home-scale utilizing Doppler effects on WiFi signals [29]. WiFinger focused on detecting finger level gestures using WiFi signals [32]. AllSee is a power-efficient gesture recognition system using TV or RFID signals [21]. These approaches have an advantage in utilizing existing wireless signals, eliminating the need for additional system deployment; however, it is unclear how robust these techniques can be in practice, since the signals will vary depending on the existing RF facilities at different places.

Finally, there are a few works using 60 GHz radar signals for gesture recognition. Soli is a miniaturized radar developed by Google for gesture recognition. They proposed a gesture recognition system with feature extraction and random forest classifier [23]. Choi et al. developed a system that recognizes gestures by applying LSTM on the range profile and Doppler profile extracted from range Doppler magnitude [7]. In contrast to these works relying on feature extraction, Wang et al. proposed a radar-based gesture recognition system closest to our work, with a model consisting of convolutional layers and an LSTM layer [35].

In this paper, we present a swipe gesture recognition technique that utilizes the Soli miniaturized radar sensor, optimized for ambient computing systems. Our work differs from existing works in many aspects. Firstly, our model is orders of magnitude more efficient than the ones presented in existing work. The model size and inference time is less than 1/5000 compared to the model proposed in [35], enabling the model to run on computationally limited devices such as mobile phones. Secondly, we evaluated our model with a more practical task where the algorithm has to recognize gestures with unsegmented time series data. In contrast, existing works [7, 23, 35] evaluated performance based on a classification task with pre-segmented data where each segment represents one and only one gesture, and gesture patterns are aligned in the time dimension. Our task is more challenging and practical, fitting the requirements for ambient computing applications outlined in the introduction of this paper. Finally, we trained and evaluated our model with significantly larger datasets, allowing higher confidence that our results can be generalized to real-world applications.

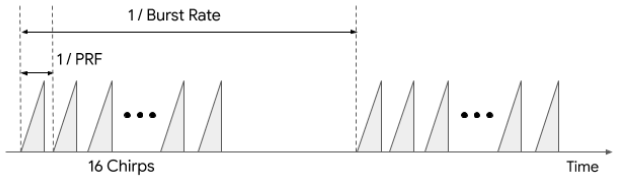


Figure 2: The radar transmits 16 chirps at PRF=2000 Hz in a burst, then stops until the next burst to save power. The bursts are transmitted at 25 Hz).

3 RADAR PRINCIPLES

Our sensing system uses the Soli frequency modulated continuous wave (FMCW) radar, based on the principles described in [23]. At a high level, we illuminate the surroundings, including hands and body, with a broad 150 degree radar beam with chirps repeated at very high pulse repetition rate (PRF). In contrast to [23], we utilize a burst transmission scheme: we send 16 chirps in a burst (Figure 2), then stop transmitting until the next burst of chirps. Using a PRF of 2000 Hz and burst rate of 25 Hz, the overall transmission duty cycle is less than 2%. This burst scheme significantly reduces the average sensor power consumption compared to transmission patterns used in existing works [23, 35] where radar chips continuously sent out chirps.

The radar's receiving antennas capture a superposition of reflections from scattering surfaces in Soli's detection range and field of view. We process the received signal from each burst with multiple stages of FFT to generate the complex range Doppler map [23], a two-dimensional representation of the reflected radar signal (Figure 11). The range dimension corresponds to the distance from a Soli sensor to a surface that reflects the radar signal. The Doppler dimension corresponds to the velocity of the reflector towards the Soli sensor. This mapping thus separates reflections that are superimposed in the fast time domain into resolvable reflections from objects at different distances or different velocities. The number of range and Doppler bins in the range Doppler map are decided by the FFT window sizes, which are set to 64 and 16 respectively in our system. The range resolution Δr and velocity resolution Δv are determined by radar system parameters:

$$\Delta r = \frac{c}{2BW} = 0.033 \text{ m} \tag{1}$$

$$\Delta v = \frac{cPRF}{2f_c l} = 0.31 \text{ m/s} \tag{2}$$

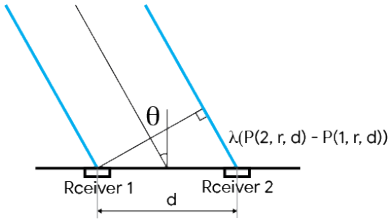


Figure 3: Blue lines denote the reflected signal from a target. The direction of arrival can be computed based on the path difference as shown in Eq. 4.

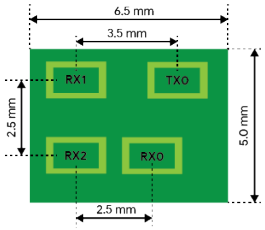


Figure 4: The second generation Soli chip has one transmitter (TX) and three receivers (RX). The receivers are positioned in an L shape with 2.5 mm gaps.

where c denotes the speed of light; BW is the transmission bandwidth, which is set to 4.5 GHz; f_c is the center frequency, which is set to 60.75 GHz; and l is the number of chirps, which is set to 16.

The absolute range Doppler map $ARD(r, d)$ (Figure 1, top) is obtained by taking the magnitude of the complex range Doppler map $CRD(r, d)$, where r and d are the range and Doppler bin indices respectively:

$$ARD(r, d) = \text{abs}(CRD(r, d)) \tag{3}$$

The absolute range Doppler value at each (r, d) bin represents the amount of reflected energy corresponding to that range and velocity. The location of peaks in the absolute range Doppler map thus indicate positions of reflectors in the range Doppler space.

The interferometric range Doppler map shows the directions of arrival for the reflected energy (Figure 1, bottom) and can be computed from the complex range Doppler maps generated by pairs of receivers. With two receivers, when we can assume that a target is far enough compared to the wavelength λ and antenna gap a, the direction of arrival θ in a plane containing two receivers and perpendicular to a chip surface can be computed as

$$\theta(r, d) = \arcsin \frac{\lambda \arg(CRD_1(r, d) CRD_2^*(r, d))}{a}, \tag{4}$$

where $CRD_i(r, d)$ is the complex range Doppler value corresponding to receiver i, and x^* denotes the complex conjugate of x (Figure 3). We designed the Soli antenna spacing to be a half of the wavelength to make it possible to compute $\theta(r, d)$ without ambiguity in $(-\pi/2, \pi/2)$, which covers Soli's field of view.

It is important to note that when multiple targets (e.g., a hand and a body) are at the same distance and velocity, the reflections from the targets appear in the same range-Doppler bin and cannot be resolved, resulting in inaccurate direction of arrival estimation in Eq. 4.

Our chip has one transmitter and three receivers in an L shape (Figure 4). We can compute two angles using two pairs of receivers: (RX0, RX2) and (RX1, RX2). Based on the range obtained from the absolute range Doppler map and the two angles, we can determine the three-dimensional positions of objects around the Soli sensor.

4 GESTURE DEFINITION

The interaction technique proposed in this paper is based on the recognition of mid-air swipe gestures. The swipe gesture resembles movements that we perform to manipulate physical objects (e.g., swiping things away), as well as digital objects on a touch-surface.

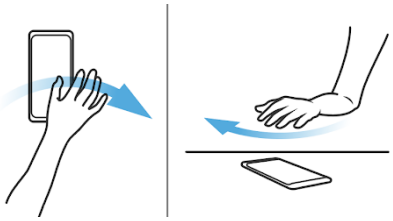


Figure 5: A swipe motion was a hand moving from one side of a phone to the other, with a part of the hand passing above the phone.

It is familiar to the user, easy to execute and memorize. As discussed in the Use Cases section of this paper, mid-air swipes are simple gestures with the potential to enable a large set of use cases.

We broadly defined swipes as sweeping motions that crossed both sides of a device, similar to brushing crumbs off a table (Figure 5). In particular, we defined five instances of the swipe gesture: four directional swipes and what we called an "omni-swipe." Directional swipes were defined as hands traveling parallel to the device display in one of the four directions (up, down, left, and right). "Omni-swipes" were defined as hands traveling in any direction parallel to the screen, including diagonal movements. Numerically, the swipes were defined as movements that satisfied the following conditions: 1) An object should move from one side to the opposite side of a device within 0.5 sec; 2) the height of the object should be higher than 3 cm and lower than 20 cm; and 3) when viewed from the direction perpendicular to the display surface, a part of the object should move over the display region.

The directionality of swipes were defined from a device-centric perspective. For instance, a hand moving from left to right when a phone was in portrait orientation was defined as a right swipe. The same motion was recognized as a downwards swipe when a phone was in landscape mode with its top on the left. Hand positions were not specified in our pilot studies: we found that some people prefer hand orientations perpendicular to the display surface while other people prefer parallel.

5 RADARNET ALGORITHM

This section presents RadarNet, a novel algorithm for gesture recognition using radars. In the current paper we apply and validate

RadarNet to recognize directional swipe gestures (i.e., up, down, left, and right swipes) at short range, as well as omni-swipes (i.e., swipes in any direction). The basic principles of the RadarNet, however, are generic and can be used to design a broad variety of novel algorithms for gesture recognition using radar sensors.

The RadarNet gesture recognition pipeline is presented in Figure 6 and consists of a) radar signal processing, b) a novel convolutional neural network architecture (Figure 7), and c) a gesture debouncer. The pipeline uses radar signals received from 25 Hz burst transmissions as inputs. In each frame, the radar chip sends a burst of 16 chirps and captures reflections from surrounding objects. Radar signal processing algorithms converts the received signals into complex range Doppler maps. The frame model of RadarNet further processes the range Doppler maps into a frame summary consisting of 32 values. The temporal model of the RadarNet combines the last 12 frames of the summaries and outputs predictions. Finally, the gesture debouncer processes the predictions to recognize the swipe gestures.

5.1 RadarNet Inputs

We use complex range Doppler maps as inputs, in contrast to existing works which used absolute range Doppler maps [7, 35]. The complex range Doppler maps contain phase information about the angular position of targets; the model can therefore recognize directional gestures better with complex range Doppler maps. We also tried using absolute range Doppler maps and interferometric range Doppler maps as inputs to the model, considering that data representations easier for humans to understand could result in better model performance; however, there were no significant differences in the model performance. Thus, we opted to use the complex range Doppler as inputs to eliminate the computational costs of processing complex range Doppler maps into absolute range Doppler and interferometric range Doppler maps. It is important to note that there was no clear evidence that the model calculated data representations similar to interferometric maps; however, passing the full complex range Doppler maps as inputs allowed the model to utilize the information when it was helpful to make predictions.

From the three receivers, we compute three complex range Doppler maps at every frame. Each map is two-dimensional complex data with 64 range bins and 16 Doppler bins. One frame is 64 range bins $\times$ 16 Doppler bins $\times$ 3 receivers $\times$ 2 values (real and imaginary) as a float representation. Because the swipe motions occur near the phones, we crop the range bins at the 24th bin, corresponding to 0.79 m. The data is reshaped into a tensor with size $24 \times 16 \times 6$ and passed to the frame model. We experimented with different shapes of input tensors and found that it was crucial to combine the complex dimension and the receiver dimension and map it to the channel of the input tensor. This is because a range Doppler cell in the tensor contains complete interferomety information representing angular position.

5.2 RadarNet Outputs

The RadarNet outputs three sets of predictions (Table 1). The portrait, landscape, and omni predictions include three, three, and two classes respectively. Within each prediction, the class probabilities sum up to 1. The sum of all probabilities across the three predictions is 3 because classes in different predictions are not mutually exclusive. The omni-swipes are defined as swipes in any direction; hence they include directional swipes. Furthermore, by making portrait predictions (left, right, background) and landscape predictions (up, down, background) independent, the algorithm can recognize diagonal swipes when both portrait and landscape predictions are triggered. Reflecting this design choice, the RadarNet has three dense layers after LSTM (Figure 7(a)). Each of the dense layers outputs probabilities in the three prediction sets.

Table 1: RadarNet outputs three predictions: portrait, landscape, and omni. These predictions contain three, three, and two class probabilities respectively. Note that the classes within a prediction are mutually exclusive, while classes between two predictions are not mutually exclusive.

Predictions	*Classes*
Portrait Prediction	Right, Left, or Background
Landscape Prediction	Up, Down, or Background
Omni Prediction	Omni or Background

5.3 RadarNet Architecture

The RadarNet consists of a frame model and a temporal model. The frame model has convolution, pooling, and activation layers utilizing residual blocks [17] and bottleneck blocks [31]. At the beginning of the model, an input tensor is sliced in the range dimension and a 1×1 residual block is applied. This is to compensate the wide dynamic range of the complex range Doppler maps in the range dimension. The 1×1 residual block is capable of doing calculations similar to interferometry. Next, the slices are concatenated, and a bottleneck block and a 3×3 residual block are applied. Finally, three convolutional layers and two dense layers are applied to summarize a tensor into 32 values. For all convolution layers, circular padding is used in the Doppler dimension to compensate for any Doppler aliasing and zero padding is used in the range dimension.

The temporal model concatenates the summary from the current frame with the summaries from the last 11 frames and passes them into a LSTM layer. The output from the LSTM layer is then passed to three dense layers with softmax, which output the three sets of class probabilities corresponding to the three predictions in Table 1.

5.4 Gesture Debouncer

RadarNet outputs class probabilities for a given segment of 12 frames, which we call a segmented classification task. In practice, however, an algorithm must recognize gestures from an unsegmented data stream. This unsegmented recognition task is significantly more difficult than the segmented classification task since it is unknown where gestures are within the unsegmented time series data. To perform the unsegmented recognition task, we added the following heuristics using predictions from the RadarNet as inputs:

- For a gesture to be detected, the likelihood of the gesture should be higher than a threshold in the last three consecutive frames.
- After one gesture is detected, all gesture likelihoods should become lower than 0.3 before the next gesture is detected.

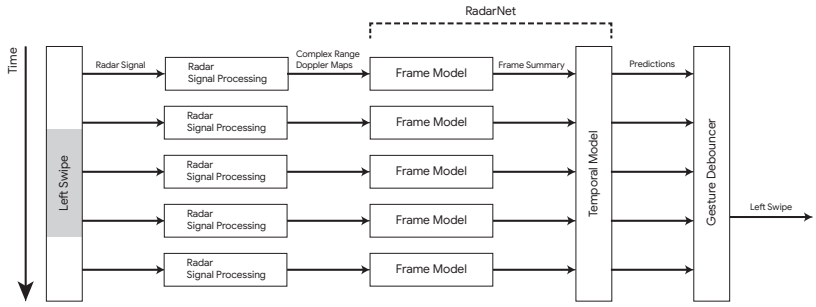
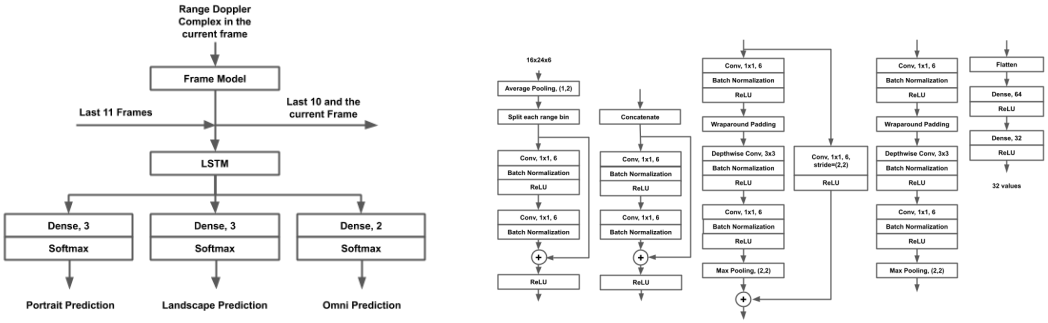


Figure 6: The pipeline applies signal processing algorithms to the radar signals to compute complex range Doppler maps for each burst. The complex range Doppler maps are converted into a summary consisting of 32 values with the frame model of the RadarNet. Then the summaries in the last 12 frames are processed by the temporal model of the RadarNet to make portrait, landscape, and omni predictions. Finally, the gesture debouncer output recognizes gestures.



(a) Temporal Model (b) Frame Model

Figure 7: RadarNet consists of a temporal model and a frame model. The frame model summarizes one frame of complex range Doppler maps into 32 values. The temporal model combines summaries from 12 frames with LSTM, then applies three dense layers that output three sets of probabilities for the portrait, landscape, and omni-swipe predictions.

The thresholds were experimentally determined in order to achieve a desired balance between recall and false positives.

6 HARDWARE

We developed a new Soli chip with one transmitter and three receivers (Figure 8). The number of antennas was decreased from two transmitters and four receivers in the previous Soli chip [23], allowing the chip's footprint to shrink from 12×12 mm to 6.5×5.0 mm. Though the antenna reduction caused a decrease in the signal to the noise ratio, we opted to adopt the new antenna configuration to make the chip's footprint small enough to be integrated into mobile phones. The receivers were aligned in an L shape with 2.5 mm gaps between antennas (Figure 4). The Soli chip was located in the top bezel of the phone (Figure 9). Since the RF signal penetrates through a plastic enclosure, no aperture was necessary on top of the chip. This benefited the industrial design of the device, as well as enabling features such as waterproofing.

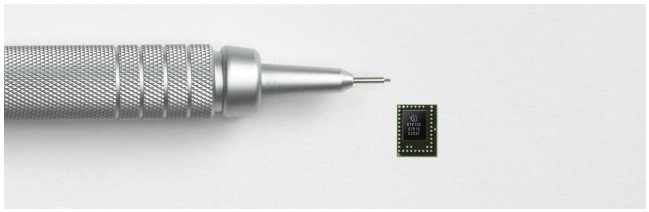


Figure 8: A full-scale photo of the new Soli chip (5.5 × 6.0 mm).

7 DATA COLLECTION

To train and evaluate our models, we collected 5019 hours of gesture samples and negative samples. This is a few hundred times larger than the datasets used in existing works (e.g, [35]) on gesture recognition with radar samples. The data collection entailed recording radar signals while participants performed swipe gestures and non-swipe motions around phones, such as reaching toward the phone and interacting with the touchscreen. The positive and negative data collections are described in detail below.

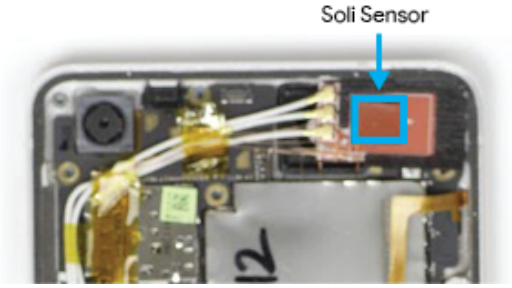
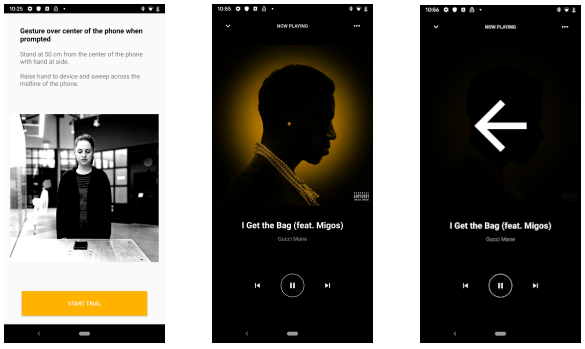


Figure 9: The Soli chip was integrated into the top of the phone. Since the radar signal penetrates though the plastic enclosure, it did not require an aperture on top of the chip.



(a) Instructions (b) Music UIs (c) Gesture Prompt

Figure 10: Positive data collection application. After the instructions were displayed (a), a music UI (b) and gesture prompts (c) were displayed.

7.1 Positive Data Collection

We collected swipe gesture samples from 7647 participants recruited from our organization at nine different locations. All participants were recruited via emails and websites in our organization. 63% of the participants were male, 36% of them were female, and 1% of them were unspecified. 86% of them were right-handed, 5.7% of them were left-handed, 4.6% of them were ambidextrous, and 3.5% of them were unspecified. The data collections were conducted by proctors who participated in two day training sessions to make the data collection as consistent as possible. Each data collection study lasted about 30 minutes. Initially, participants were asked to read and sign consent forms. They were then shown instruction slides explaining the overall procedure of the data collection session, data collection UI, and videos of swipe gestures from two different angles. In our pilot studies, we gave the quantitative definition of the swipe motions to participants, assuming that it would help the participants perform gestures satisfying the conditions. However, this resulted in unnatural robotic swipe motions because participants performed gestures too carefully given the detailed instructions. We therefore opted to give visual instructions instead for this study, and, later, filter out the gestures that did not satisfy the numerical conditions from the dataset.

After watching the slides, participants were instructed to interact with a phone placed on a desk. All ensuing instructions were displayed on the phone. To maximize the quantity of gesture samples collected, participants were asked to complete as many sessions as possible depending on how much time was left after all instructions were given. In each session, we collected 12 gesture samples in an experimental condition. The conditions were combinations of 1) the participant's postures (sitting and standing), 2) the phone's placements (on a desk and in the participant's hand), and 3) the phone's orientation (portrait, landscape with the top of phone at participant's right, and landscape with the top of a phone at participant's left). A combination of these parameters was randomly chosen and displayed on the phone. The probabilities for the orientation setting were set to balance the numbers of gestures collected in portrait and landscape orientations.

Once the participant pressed the start trial button (Figure 10(a)), a UI emulating a music application was displayed (Figure 10(b)) and the application started radar signal recording. Then an arrow appeared at the center of the display indicating the directions of swipes (Figure 10(c)). The participant was instructed to perform swipe gestures in the directions indicated by the arrows. After the gesture prompt and execution, the participant was given distraction tasks, such as picking up an object near the device (e.g., a pen), moving an object over the device (e.g., a cup), reaching and picking up the device, and tapping on the screen, in order to avoid habituation. Four directions (up, down, left, and right) were prompted three times in a randomized order for each session.

When the participant performed each gesture, a proctor pressed a button on a remote clicker to provide visual feedback on the phone's display, as if the phone had responded to the participant's swipe. The time window from the gesture prompt to the button click was labeled as a gesture segment and recorded with a swipe direction. The labels were used to create datasets and perform evaluations.

7.2 Negative Data Collection

We collected radar signals while people performed motions similar to swipe motions for our machine learning training as well as algorithm evaluations. We recorded 285 hours of data in which participants interacted with phones in various ways, such as reaching towards the phone and interacting with the touchscreen.

To collect data in a wider variety of contexts, we also experimentally defined the following five categories: 1) walking tasks with phone in hand, 2) common movement tasks near the phone, 3) common movement tasks when phone is in hand, 4) common movement tasks when phone is laying flat, and 5) common movement tasks (in X, Y, Z planes) when a user interacts with their phone. We defined about 10 scenarios for each category for a total of 52 scenarios. Each scenario falls into one of the two types: 1) recordings of natural behaviors around a phone that include motions similar to swipes (e.g., wiping a table with a phone on the table) and 2) repetitions of a hand movements similar to swipe gestures (e.g., moving an object from one side of the phone to the other side). In existing works, systems had been trained and evaluated with the first type; however, we opted to train and evaluate against adversarial recordings of both types to improve the robustness of our system.

Table 2: The number of positive samples for each dataset. Samples were split on recording basis. Thus, gestures from any one participant were not divided into multiple datasets.

Datasets	*Gesture Recordings*
Training	368×10^3
Development	85.5×10^3
Test	104×10^3

We asked our engineering team to perform these 52 scenarios for data collection since it was difficult from IRB perspective to let participants follow these scenarios as user studies. Data was collected with an Android application that was specifically tailored to the negative protocols defined. When the participating team member chose one of the 52 scenarios, a protocol for the scenario was shown on the phone display, such as "Place the phone in an armband and run on a treadmill for 10 minutes." During these sessions, the participant was given the following guidelines: 1) do not perform any intentional swipes, 2) perform tasks with the most natural behavior possible, 3) natural variations of laying/sitting positions are allowed to add diversity to the data set. The Android application recorded radar signals while participants followed the protocols.

7.3 Training, Development, and Test Sets

We removed invalid gesture recordings resulting from system problems or participants not following instruction. We further filtered out gestures performed too far from the display surface or with insufficient amplitude. After the data cleaning, we had 5.58×10^5 gesture recordings. We split the recordings into training, development, and test sets (Table 2). The training set was used to train machine learning models, while the development set was used to evaluate the performance of the model during training and hyper parameter tuning. The test set was used to evaluate the performance of the trained models as described in the evaluation section. The data split into these sets was done on a user study basis; thus, data from the same participant was not split into multiple data sets.

After the data split, we refined labels attached in the positive data collection. Because there were delays between the time when the arrows were displayed and when participants started performing gestures, as well as from the time when the participants performed gestures and when proctors clicked the button, we post-processed the labels with a label refinement algorithm to identify center frames where participants' hands were closest to the Soli sensors. Twelve frames around the center frame were extracted as a positive sample for each gesture. For the training set, we also extracted two time windows before and after the window to increase the variability in the time domain. We also augmented the positive samples by three times by scaling the radar signal with a factor randomly chosen from a normal distribution with a mean of 1 and a standard deviation of 0.025.

We generated negative samples by extracting 12 frames at a fixed time interval from the negative recordings, then split into the training, development, and test sets. Since we had an excessive number of negative samples compared to the number of positive samples, we decimated negative data to keep the ratio of positive to negative samples at 1:6, which was chosen experimentally to optimize the model performance.

To each sample, we attached three labels: a portrait swipe label, a landscape swipe label, and an omni-swipe label corresponding to the three predictions in Table 1. Positive samples had at least one of the labels other than background. Negative samples had all labels as background.

The evaluation set was also used as time series data without segmentation for the unsegmented recognition task described in the Evaluation section. In this task, time windows where participants performed gestures were labeled with the type of gestures based on the label attached in the positive data collection. Other frames were labeled as background.

7.4 Samples

To better understand the sensor data, we visualized some of the samples in our dataset. Figure 11 shows an example of a clean left swipe visualized as a series of complex range Doppler maps. Each column is a time frame from left to right. The top two rows are real and imaginary values from RX0, the next two rows are from RX1, and the bottom two rows are from RX2.

We also visualized absolute range Doppler maps (Eq. 3) and interferometric range Doppler maps along with the swipe direction (Eq. 4) to derive some insight into the signal (Figure 1), using the same gesture sample as in Figure 11. In these maps, the X axis corresponds to the velocity toward the sensor. The velocity is zero at the center, negative on the left side, and positive on the right side. The Y axis denotes range (i.e., the distance from the sensor). The range is smallest at the bottom and largest at the top. The velocity resolution and range resolution are 0.31 m/s and 0.033 m, as shown in Eq. 1 and 2. In Figure 1 top, we see a peak (i.e., a user's hand) that approached the sensor from the first to the sixth frames and moved away after that. Additionally, in the interferometric maps, we see that the cells corresponding to the range Doppler peak change their colors from red to green to blue, indicating that the angle of the hand changed as a user performed a left swipe motion. Figure 12 also shows positive samples. The top is a clean right swipe, illustrating that the pattern in the absolute range Doppler maps is very similar to the one in the left swipe sample, while the interferometric map color changes in the opposite order. These left and right swipe samples had clean motion signatures; however, many of the actual positive samples showed less ideal patterns. The middle rows correspond to a small left swipe with subtle changes in the absolute range Doppler and interferometric maps. The bottom shows a left swipe with a user's body near a phone, making the motion pattern in the interferometric maps unclear.

Figure 13 shows examples of negative samples with patterns similar to swipe motions. Moving a hand up and down (top) creates similar patterns in the absolute range Doppler maps because the hand approached to the sensor and then moved away. However, because the hand's angle did not change much, the color of the interferometric map bins corresponding to the range Doppler peak does not change, though noise in the interferometric maps makes it difficult to distinguish this motion from swipes with small amplitudes. The touchscreen swipes (middle) and picking up an object

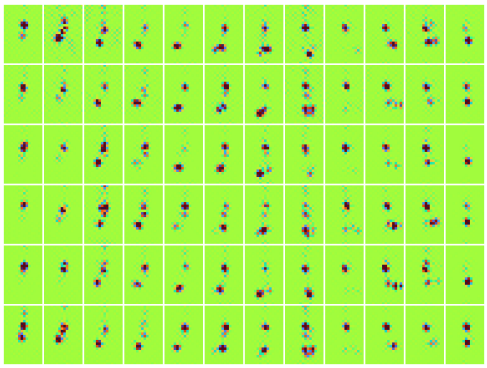


Figure 11: An example of a clean left swipe motion. Rows represent real and complex components of range Doppler data from three channels. Columns represent time series of 12 frames. In each heat map, the X axis denote velocity, with 0 at the center, negative velocities on the left side, and positive velocities in the right side. The Y axis denotes the distance from the sensor, increasing from bottom to top. The majority of cells are green, indicating values near 0. We can see an object moving close to the Soli chip around the sixth frame, then moving away.

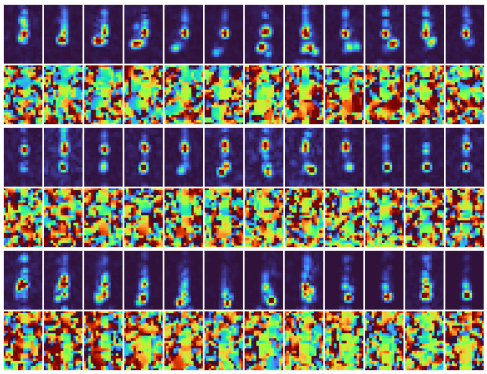


Figure 12: Positive sample visualizations of a clean right swipe (top), a small left swipe (middle), and a left swipe with the user's body near a phone (bottom). Many of our positive samples were similar to the one in the middle or bottom without clean patterns in the data, making the classification challenging.

near a phone (bottom) also have patterns similar to the in-air swipe motions.

Such factors, including but not limited to unclear gesture patterns, similarities between positive and negative samples, and low signal to noise ratios in the interferometric maps, make it challenging for an algorithm to robustly recognize swipe motions.

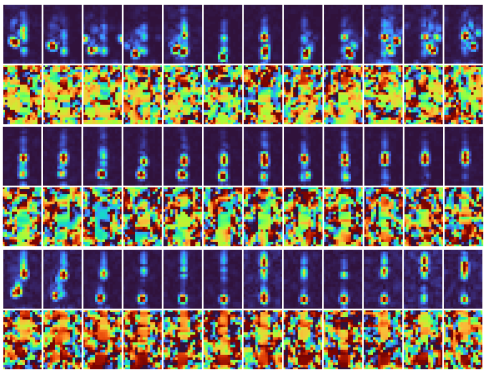


Figure 13: Negative samples visualization of a hand moving up and down (top), a touchscreen swipe (middle), and picking up an object near a phone (bottom). Many motions including these examples created similar patterns as swipes.

8 EVALUATIONS

We trained RadarNet using the training set. All training was done for 1.5×10^6 steps with a batch size of 128. Each training took about five to six hours using our servers. We evaluated the performance of our algorithm in a segmented classification task and an unsegmented recognition task. In the segmented classification task, the model had to classify segmented data into classes, while in the unsegmented recognition task, the model had to recognize gestures from unsegmented time series data. Since it was unknown where gestures existed in the time series data, the unsegmented recognition task was more difficult. However, models must process unsegmented time series data in practical applications; the unsegmented recognition task thus gives more ecologically valid performance estimates than the segmented classification task. Furthermore, in the segmented classification task, our dataset contains challenging non-gesture samples with radar signal patterns similar to gestures, while in existing works (e.g., [7, 23, 35]), samples in the dataset always represented one gesture with distinct motion patterns from other gestures in their dataset. The inclusion of challenging non-gesture samples made our segmented classification task more ecologically valid and difficult. In the following, we evaluate the proposed algorithm with the segmented classification task, then with the unsegmented recognition task.

8.1 Segmented Classification Task

We used the test set pre-segmented into samples each containing 12 frames of complex range Doppler maps. Each sample had three labels: a portrait swipe label, a landscape swipe label, and an omni-swipe label. The segmented classification task is for a model to predict one class for each of the three predictions (Table 1) to match to the ground truth labels attached to samples, given 12 frames of radar signals. Table 3 shows the accuracy of each prediction at the points where recall and precision were equal. All prediction accuracies were higher than 0.99. Figure 14 shows the precision-recall curve for each prediction. The precision and recall were averaged excluding background in each prediction. The performance curves

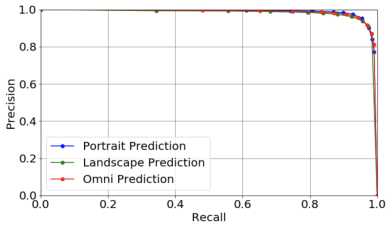


Figure 14: The precision-recall curves of the portrait, landscape and omni predictions indicate that RadarNet provides robust performance in the segmented classification task.

Table 3: The accuracy is above 0.99 for all classes, showing that the RadarNet achieves robust performance in the segmented classification task.

Prediction	*Class*	*Accuracy*
Portrait Prediction	Right	0.9943
	Left	0.9924
	Background	0.9945
Landscape Prediction	Up	0.9928
	Down	0.9937
	Background	0.9948
Omni Prediction	Omni	0.9930
	Background	0.9929

are close to the top right corner. These results demonstrate that RadarNet provides robust classification performance in the segmented classification task.

8.2 Computational Efficiency

We assessed the efficiency of the RadarNet based on its model size and inference time. The model size affects how much memory is required to run the model. Inference time affects how much computational power a processor needs to run the model, as well as the power consumption of the computation. Therefore, these metrics are of greatest importance to assess the model's efficiency. We used two models as baselines: the end-to-end model proposed in [35] as a gesture recognition model with 60 GHz radar signals and MobileNet [19] as a model designed specifically for mobile devices.

We evaluated inference time using the TensorFlow Lite [10] performance profiler. We converted all models used into TFLite models and measured the inference time on Pixel 4 XL by taking the average inference times over 5000 runs. As shown in the Table 4, RadarNet was significantly more efficient compared to other models. Its model size and inference time were factors of 2.0×10^{-4} and 1.1×10^{-6} smaller than the model size and inference time in [35]. Although the model in [35] recognized 11 gestures while our model recognized five gestures, these efficiency differences are significant. As described in the Model Structure section, RadarNet cached the output from its frame model to reduce the computation. The end-to-end model did not adopt a similar caching technique although it was technically feasible. Thus, as another reference point, we evaluated the inference time of the RadarNet without the caching.

Table 4: Comparison of computational efficiency in terms of model sizes and inference times. RadarNet was significantly more efficient compared to other models.

Model	*Model Size [MB]*	*Inference Time [ms]*
RadarNet	0.14	0.147
RadarNet w/o caching	0.14	0.909
End-to-end [35]	689	1.37×10^5
MobileNet [19]	17	13.2

Removing the caching increased the inference time of the RadarNet to 0.909 ms. However, it was still significantly smaller than that of the end-to-end model in [35].

Since the end-to-end model was not designed to be executed on mobile devices, we also compared our model with MobileNet [19]. MobileNet is an image classification model specifically designed for mobile devices and thus is appropriate as another baseline. RadarNet's model size and inference time are 0.8% and 11% respectively of MobileNet's. These comparisons to the MobileNet baseline demonstrate that RadarNet is small enough to be executed on mobile devices with limited computational resources and power.

8.3 Unsegmented Recognition Task

We also evaluated our algorithm with the unsegmented recognition task. In this task, the input to the algorithm was radar signal time series, as opposed to pre-segmented samples in the segmented classification task. We used recordings in the test set without segmentation.

We used the recordings from the positive data collection to calculate the accuracy. The recordings had labels, attached during data collection, indicating time windows where participants performed gestures. We marked a time window *correct* if the algorithm output one and only one correct prediction in the time window. We marked the time window *wrong* if there were no predictions, a wrong prediction(s), or multiple correct predictions. Then, we calculated the unsegmented detection rate by dividing the number of *correct* time windows by the total number of time windows. There were 1.04×10^5 gesture time windows in the recordings.

We used a subset of recordings in the test set from the negative data collection to calculate false positives per hour. As described in the Negative Data Collection section, we collected two types of negative data: 1) natural behaviors around a phone that include hand motions similar to swipes (e.g., wiping a table with a phone on the table) and 2) repetitive hand movements similar to swipe gestures (e.g., moving an object from one side of a phone to the other side). We used the first type in this analysis to assess the system's performance in practical scenarios. The data in the second type were used in the next section where we evaluated the system's robustness against an adversarial dataset. Please note that we performed this evaluation using recordings of behaviors that were likely to cause false positives. Thus, the numbers of false positives per hour reported in this section are close to the upper bound rather than the expected value. We used false positives per hour as a metric because there were no clear definitions of the number of gesture-like motions in the negative recordings. The total length of recordings was 40.2 hours.

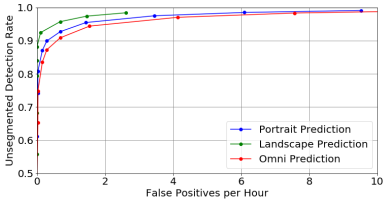


Figure 15: With recordings of natural behaviors around phones as negative data, the false positives per hour for the portrait, landscape, and omni predictions were 0.03, 0.0, and 0.06 respectively when the unsegmented detection rate was 0.8. These numbers show that the system is robust enough to always-on in practical contexts.

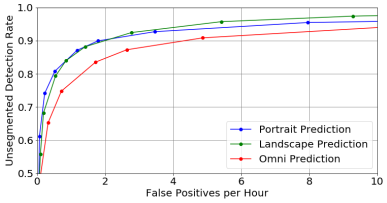


Figure 16: With recordings of repetitive hand motions similar to swipe gestures, the false positives per hour for the portrait, landscape, and omni predictions were 0.5, 0.5, and 1.0 respectively when the unsegmented detection rate was 0.8. The number of false positives is small considering that swipe-like motions were repeated every few seconds in these recordings.

The detection rate and the false positives per hour are a function of the threshold value used in the gesture debouncer. Figure 15 shows graphs of the detection rates and the false positives per hour for the portrait, landscape, and omni predictions. When the unsegmented detection rate was 0.8, the false positives per hour for the portrait, landscape, and omni prediction were 0.03, 0.0, and 0.06 respectively. These results indicate that the false positive rate of our algorithm is small enough for the system to be always-on, minimizing frictions of using gesture-based interaction in practical contexts.

8.4 Robustness against Adversarial Dataset

We further evaluated our system against an adversarial dataset trying to break our system. The adversarial dataset was derived from negative recordings of repetitive hand motions similar to swipe gestures. For example, the dataset included recordings of people moving an object from one side of a phone to the other side, touching a screen, doing a screen swipe up gesture (i.e., Android's phone unlock gesture), and turning a car's steering wheel with a phone in a holder placed nearby. Figure 16 shows the graph of detection rates and false positives per hour for the portrait, landscape, and omni predictions. When the unsegmented detection rate was 0.8, the false positives per hour for the portrait, landscape, and omni predictions were 0.5, 0.5, and 1.0 respectively. Since these recordings consisted of people doing these tasks repetitively every few seconds, the recordings contain hundreds of motions per hour. Considering this, the numbers of false positives were very small. In practice, users are unlikely to repeat these motions for hours continuously. Therefore, we believe that the results indicate our system is also robust against motions very similar to swipe gestures.

8.5 Performance and Dataset Size

When we developed a machine learning-based gesture recognition technique, one of the biggest questions was how many gesture samples we should collect. We collected over 5.0×10^5 gesture samples in this work, which gave us the opportunity to evaluate the relationship between the number of training samples and performance of the model at scale, which had not been previously reported in our community.

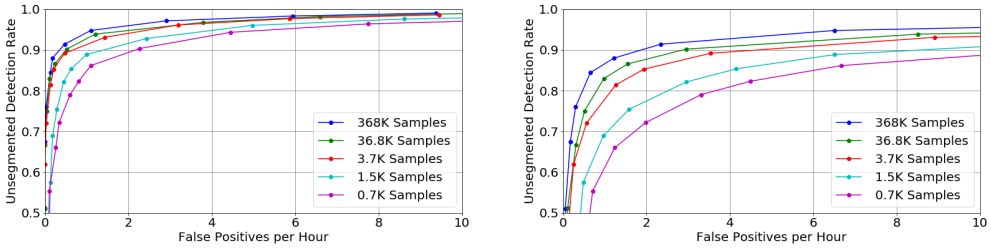
To investigate the effect of the number of gesture recordings used to generate the training sets, we generated multiple datasets using different numbers of gestures recordings and evaluated the change in performance. Each training set was generated with 0.7×10^3 to 368×10^3 gesture recordings. We extracted three positive samples per gesture recording and augmented them three times as described in the Train, Development, and Test Sets section. The negative samples were randomly sampled to keep a 1:6 ratio of positive to negative samples. We trained RadarNet using the datasets and computed the average accuracy of the portrait predictions and false positives per hour (Figure 17). The graphs show that the increase in data sizes provided bigger performance improvements against the adversarial dataset than against the natural dataset, indicating that we needed more samples to discriminate more difficult samples. Another interesting observation is that a model trained with 3.7×10^3 gesture recordings could achieve 90% or more of the performance of the model trained with 3.68×10^5. This implies that we can assess the performance of gesture recognition techniques with deep learning reasonably well with much smaller datasets.

It is important to note that we randomly sampled recordings for the smaller datasets from all recordings; hence the small datasets still reflected the variability in the full dataset. If the size of data collection is reduced without considering the variability, the performance could be overestimated. For example, if we collect gesture recordings from a small number of participants, models trained with the recordings would overfit to the dataset, showing high performance but not generalizing to the general population. Thus we should interpret the number of recordings suggested in this analysis as lower bounds rather than target numbers to have a certain amount of confidence in our performance estimates.

9 USE CASES

The swipe detection technique introduced in this paper opens the opportunity to design novel interactions for mobile devices that allow users to quickly manage interruptions, accelerate common tasks, and manage multi-tasking when touch-based interactions are inconvenient or not available.

Phone calls, alarms, and notifications often cause frustration if users cannot deal with them immediately. On mobile devices,



(a) With Natural Negative Dataset

(b) With Adversarial Negative Dataset

Figure 17: The detection rate-false positive per hour graph for the portrait prediction changed with the number of positive recordings used to generate training sets. Interestingly, with 1% of total positive samples (i.e., 3.7×10^3 samples), the model achieved 90% of the accuracy obtained by a model trained with all positive samples (i.e., 3.68×10^5 samples)

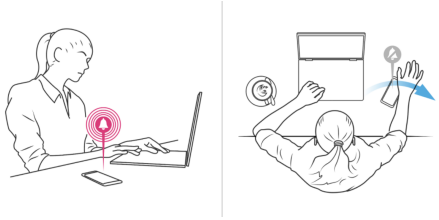


Figure 18: Mute a call by swiping near a phone without interrupting a primary task.

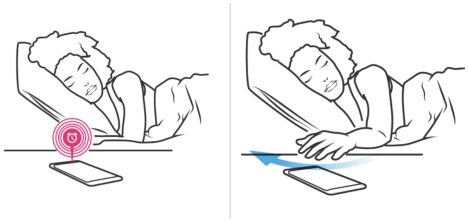


Figure 19: Snooze an alarm by swiping near a phone without fine eye-hand coordination.

we rely on UI elements and touch interactions to manage these interruptions. However this modality requires fine eye-hand coordination that is often undesirable for situations which require immediate action; for example, when driving a car, attending a meeting, or waking up in the morning. The swipe gesture removes this friction with a simple eyes-free movement to handle interruptions (Figure 18 and 19).

The swipe gesture can also be used to accelerate common and frequent tasks, such as skipping tracks while listening to music. Currently, with mobile phones, users may need to unlock the device, find the music controls, and then press the skip button. Always-on gesture recognition can be used to execute predefined functionalities with a single motion, regardless of the device state. This allows the user to move to the next or previous song by swiping left or right, regardless of whether the mobile phone is locked or the music application is in the background (Figure 20).

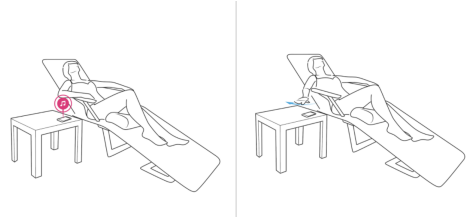


Figure 20: Always-on gesture recognition technique allows users to execute predefined functionalities, such as skip music, with swipe gestures.

In certain contexts, touching the device screen is inconvenient or not desirable. For example, while we are cooking, we might want to follow a recipe on our phone when our hands are dirty. The recognition of a touchless swipe gesture allows users to quickly move from reading the next step of the recipe on the screen to executing the instructions on the kitchen counter without cleaning their hands in between.

Given its small footprint, low compute requirement, low power consumption, and privacy preserving capabilities, the Soli sensor is uniquely positioned to fit the strict technological requirements of ambient computing applications. A swipe gesture recognition technique powered by Soli can be used in a variety of ambient computing products, supporting use cases similar to the ones described for a mobile device. For example, the user can perform a swipe gesture to manage interruptions with alarm clocks, timers, or smoke detectors. Users can also accelerate interactions with voice-activated speakers; without using hot-words to wake-up the device, the user can quickly change tracks when voice input is not desirable (e.g., when having conversation with someone or when music is too loud). In cars, while users are engaged in cognitively demanding tasks such as driving, an eyes-free swipe gesture can be used to manage contextually relevant information such as accepting or declining a change in a proposed GPS route. In public contexts, we can imagine controlling automatic doors, elevators, or faucets without touching shared surfaces due to reasons of hygiene.

10 LIMITATIONS

In this paper, we proposed a gesture recognition technique based on radar signals. The number of gestures that our technique can recognize is currently limited to five. Although we believe that it can recognize more gestures if necessary, further investigation is needed to confirm this.

Another limitation could be ecological validity of our dataset. Although we collected a large amount of gesture data, all participants were recruited within our organization. Thus the dataset reflects potential biases in the population. For instance, the algorithm may not work well for elderly people or people with motor disabilities, which could limit application of the proposed technique in the accessibility domain.

In our evaluation, we investigated how the dataset size affected performance of our model. While we believe that this analysis provides a useful data point to our community, the performance relationship is also affected by other factors, such as sensing modalities, complexity of gesture motions, and gesture sets.

11 CONCLUSION AND FUTURE DIRECTIONS

We introduced a novel swipe gesture recognition technique that satisfies five requirements for gestural interactions in ambient computing contexts. The proposed technique recognizes swipes using **privacy preserving** radar signals. Our radar sensing chip has an extraordinary **small physical footprint** of 6.2 x 5.0 mm that can be **placed behind enclosures**, making it possible to be integrated into devices with tight form factor constraints without affecting their aesthetic. Our experimental results demonstrated that our machine learning model, RadarNet, is efficient enough to be **continuously executed** on devices with limited computational resources while also providing **reliable** gesture recognition performance with practical tasks. These results demonstrated that radar-based gestural input is a promising candidate for interacting with ambient computing system. It opens the opportunity to investigate gestural interactions for devices with different functionalities, form factors, and strict technological constraints.

Although this work focused on recognition of categorical gestures, given Soli's sensitivity to motions, it is possible to extract more nuanced information from gestures. As part of our nonverbal communication repertoire, gestures express emotions and subtle meaning through properties such as speed, amplitude, and rhythm. If devices can recognize these properties, they can adapt their behaviors in a way that is richer than simple command-and-response interactions. As a simple example, an alarm clock can change how long to snooze the alarm based on the amplitude of the user's swipe.

Furthermore, we could go beyond detecting explicit gestures. In gesture-based interactions, users perform gestures explicitly as direct commands to control the behavior of a product or system; however, it could also be interesting to detect and understand users' implicit body cues. If a system can recognize implicit body cues that users perform unconsciously around the device, the system can respond even before the user starts the interaction, anticipating user's intention. For instance, when users start interacting with a mobile phone, they naturally reach towards the device. A reach is a motion cue that indicates the user's intention to start the interaction with the device. We can use this cue to proactively adapt the behavior of the system; for example, when the alarm goes off, we can progressively reduce its volume as the user reaches for the device. Other example cues include but are not limited to leaning to, turning towards, and approaching devices. If devices can anticipate users' intentions by understanding implicit body cues, we can make the human-device interaction more natural and fluid.

RF sensing with pico-radars is a new, exciting sensing approach with unique properties such as privacy preservation, low power consumption, robustness against light, high sensitivity to motions, and working through materials. Radars have now become broadly available to the HCI community (e.g., XENSIV[20]); however, using them remains very challenging and there has been a relatively small amount of HCI work on radars. We believe that our work demonstrated how radars can be used to design fundamental interactions, demystified RF/radar sensing by providing core principles for designing machine learning-based gesture recognition systems, and proposed an efficient deep neural network architecture that can be a starting point for our community. We believe these contributions will inspire researchers to explore this emerging technology.

REFERENCES

[1] Amazon. 2014. Echo. https://www.amazon.com/smart-home-devices/b?ie=UTF8&node=9818047011. Retrieved September 10, 2020.

[2] Apple. 2018. Siri. https://www.apple.com/siri/. Retrieved September 10, 2020.

[3] Richard A Bolt. 1980. "Put-that-there" Voice and gesture at the graphics interface. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*. 262–270.

[4] Alex Butler, Shahram Izadi, and Steve Hodges. 2008. SideSight: multi-" touch" interaction around small devices. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*. 201–204.

[5] Kelly E Caine, Arthur D Fisk, and Wendy A Rogers. 2006. Benefits and privacy concerns of a home equipped with a visual sensing system: A perspective from older adults. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 50. SAGE Publications Sage CA: Los Angeles, CA, 180–184.

[6] Ting-Han Chen, Sok-Ian Sou, and Yinman Lee. 2019. WiTrack: Human-to-Human Mobility Relationship Tracking in Indoor Environments Based on Spatio-Temporal Wireless Signal Strength *(Proceedings of the IEEE International Conference on Dependable, Autonomic and Secure Computing, Vol. 00)*. 788–795.

[7] Jae-Woo Choi, Si-Jung Ryu, and Jong-Hwan Kim. 2019. Short-range radar based real-time hand gesture recognition using LSTM encoder. *IEEE Access* 7 (2019), 33610–33618.

[8] Gabe Cohn, Daniel Morris, Shwetak Patel, and Desney Tan. 2012. Humantenna: using the body as an antenna for real-time whole-body interaction *(Proceedings of the SIGCHI Conference on Human Factors in Computing Systems)*. 1901–1910.

[9] Facebook. 2012. Oculus. https://www.oculus.com/. Retrieved September 10, 2020.

[10] Google. 2015. TensorFlow Lite. https://www.tensorflow.org/lite. Retrieved September 10, 2020.

[11] Google. 2016. Google Assistant. https://assistant.google.com/. Retrieved September 10, 2020.

[12] Google. 2016. Nest Home Hub. https://store.google.com/product/google_nest_hub. Retrieved September 10, 2020.

[13] Google. 2017. Nest Thermostat. https://nest.com/. Retrieved September 10, 2020.

[14] Google. 2019. Jacquard. https://atap.google.com/jacquard/. Retrieved September 10, 2020.

[15] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. 2012. Soundwave: using the doppler effect to sense gestures *(Proceedings of the SIGCHI Conference on Human Factors in Computing Systems)*. 1911–1914.

[16] Chris Harrison and Scott E Hudson. 2009. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices *(Proceedings of the 22nd annual ACM symposium on User interface software and technology)*. 121–124.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[18] Marigo Heijboer, Elise Hoven, Bert Bongers, and Saskia Bakker. 2016. Facilitating Peripheral Interaction: Design and Evaluation of Peripheral Interaction for a Gesture-Based Lighting Control with Multimodal Feedback. 20, 1 (2016).

[19] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv*

preprint arXiv:1704.04861 (2017).

[20] Infenion. 2016. Infenion XENSIV. https://www.infineon.com/cms/en/product/sensor/radar-sensors/. Retrieved September 10, 2020.

[21] Bryce Kellogg, Vamsi Talla, and Shyamnath Gollakota. 2014. Bringing gesture recognition to all devices. In *11th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 14)*. 303–316.

[22] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. 2012. Computer Vision – ECCV 2012, 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI. (2012), 852–863.

[23] Jaime Lien, Nicholas Gillian, M. Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. 2016. Soli: ubiquitous gesture sensing with millimeter wave radar. *ACM Transactions on Graphics (TOG)* 35, 4, 142.

[24] Microsoft. 2010. Kinect. https://developer.microsoft.com/en-us/windows/kinect/. Retrieved September 10, 2020.

[25] Microsoft. 2019. Hololens: Handtracking. https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/Input/HandTracking.html. Retrieved September 10, 2020.

[26] GRS Murthy and RS Jadon. 2009. A review of vision based hand gestures recognition. *International Journal of Information Technology and Knowledge Management* 2, 2 (2009), 405–410.

[27] I. Nasr, R. Jungmaier, A. Baheti, D. Noppeney, J. S. Bal, M. Wojnowski, E. Karagozler, H. Raja, J. Lien, I. Poupyrev, and S. Trotta. 2016. A Highly Integrated 60 GHz 6-Channel Transceiver With Antenna in Package for Smart Sensing and Short-Range Communications. *IEEE Journal of Solid-State Circuits* 51, 9 (2016), 2066–2076. https://doi.org/10.1109/JSSC.2016.2585621

[28] Antti Pirhonen, Stephen Brewster, and Christopher Holguin. 2002. Gestural and audio metaphors as a means of control for mobile devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 291–298.

[29] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. 2013. Whole-home gesture recognition using wireless signals *(Proceedings of the 13th Annual International Conference on Mobile Computing and Networking)*. 27–38.

[30] 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.

[31] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. 2016. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261* (2016).

[32] Sheng Tan and Jie Yang. 2016. WiFinger: leveraging commodity WiFi for fine-grained finger gesture recognition *(Proceedings of the 17th International Symposium on Mobile Ad Hoc Networking and Computing)*. 201–210.

[33] S. Trotta, D. Weber, R. W. Jungmaier, A. Baheti, J. Lien, D. Noppeney, M. Tabesh, C. Rumpler, M. Aichner, S. Albel, J. S. Bal, and I. Poupyrev. 2021 (to appear). SOLI: A Tiny Device for a New Human Machine Interface. *IEEE International Solid-State Circuits Conference* (2021 (to appear)).

[34] Radu-Daniel Vatavu. 2012. Nomadic gestures: A technique for reusing gesture commands for frequent ambient interactions. *Journal of Ambient Intelligence and Smart Environments* 4, 2 (2012), 79–93.

[35] Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev, and Otmar Hilliges. 2016. Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 851–860.

[36] Mark Weiser and John Seely Brown. 1997. The coming age of calm technology. In *Beyond calculation*. Springer, 75–85.

[37] Eric Zeng, Shrirang Mare, and Franziska Roesner. 2017. End User Security and Privacy Concerns with Smart Homes. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*. USENIX Association, Santa Clara, CA, 65–80.

[38] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. 2020. MediaPipe Hands: On-device Real-time Hand Tracking. *arXiv preprint arXiv:2006.10214* (2020).