Article:
Zhai, S, Ye, G, Tang, Z et al. (2021) Towards practical 3D ultrasound sensing on commercial-off-the-shelf mobile devices. Computer Networks, 191. 107990. ISSN: 1389-1286

https://doi.org/10.1016/j.comnet.2021.107990

Towards Practical 3D Ultrasound Sensing on Commercial-Off-the-Shelf Mobile Devices

Shuangjiao Zhai[a,b], Guixin Ye[a,b,*], Zhanyong Tang[a,b,*], Jie Ren[c], Dingyi Fang[a,b], Baoying Liu[a,b], Zheng Wang[d]

[a]*School of Information Science and Technology, Northwest University, Xi'an 710127, China*
[b]*Shaanxi International Joint Research Centre for the Battery-Free Internet of Things, Xi'an 710127, China*
[c]*School of Computer Science, Shaanxi Normal University, Xi'an 710062, China*
[d]*School of Computing, University of Leeds, Leeds LS2 9JT, U.K.*

Abstract

Ultrasound based contactless sensing has the potential to extend the interactive range of mobile devices significantly. However, existing approaches either require multiple transceiver pairs that are not available on commercial mobile devices or can only recognize simple actions with one single stroke. These drawbacks significantly limit the practicability of prior work. This article presents UltraScr, an ultrasound sensing system that can recognize the sophisticated gestures with multiple strokes (e.g., writing a capital letter) using just a single sound transceiver pair. To do so, UltraScr first uses the frequency attenuation profile (FAP) to capture the subtle hand movements. It then employs a convolutional neural network (CNN) to extract the subtle hand movements' discriminative features to build an accurate ultrasound sensing system. We go further by exploiting the rejection classification method (RCM) and incremental learning to improve the robustness of our sensing system in the end-user environment. We evaluate UltraScr by applying it to gesture recognition across different scenarios and users. Extensive experimental results show that while using only one transceiver pair, the performance of UltraScr is comparable to the multi-transceiver-paired implementations. We show that UltraScr is robust to the change of external conditions (i.e., different humidity and battery) and can work effectively on a wide range of locations, but requires 3× fewer training samples compared to the state-of-the-art.

Keywords: Wireless Sensing, Frequency Attenuation Profile, Ultrasound Sensing, Machine Learning

1. Introduction

Gestures are a natural way of human-computer interactions. Indeed, gesture recognition underpins many mobile applications and is emerging as a vital technology for new

*Corresponding author
Email addresses: sjzhai@stumail.nwu.edu.cn (Shuangjiao Zhai), gxye@nwu.edu.cn (Guixin Ye), zytang@nwu.edu.cn (Zhanyong Tang), renjie@snnu.edu.cn (Jie Ren), dyf@nwu.edu.cn (Dingyi Fang), paola.liu@nwu.edu.cn (Baoying Liu), z.wang5@leeds.ac.uk (Zheng Wang)

application domains like augmented reality or virtual reality [1]. Traditionally, gesture recognition on mobile devices is achieved through a touch device (e.g., a touch screen). Due to the small-form-factor of mobile devices, gesture recognition is restricted to finger movements on mobile devices. However, finger movements are just one type of gesture. There are many other hand movement gestures that people perform daily. If we can extend the reach of gesture recognition on mobile devices to such gestures, we can then unlock many new applications.

The success of vision-based gesture recognition systems like the `Xbox Kinect` is an excellent case of what could be achieved by supporting a more extensive range of gestures. For example, by recognizing a hand motion in the air, one can dim the light or change the music player's volume. Such a capability is particularly useful in application domains like smart homes, health care, and gaming. In recent years, wireless sensing is emerging as a viable means for supporting gesture recognition. Compared to a vision-based solution [2–7], wireless-based approaches have the advantages of not requiring instrumenting the users and being less privacy intrusive [8]. Among many wireless media like WiFi [9–15] and RFID [16–19], sound waves [20–28] are particularly interesting in mobile systems because most commercial off-the-shelf mobile devices already have a sound transceiver pair (i.e., a speaker and microphone).

While promising, existing approaches have significant drawbacks. Approaches like LLAP [20] can barely recognize elaborate gestures with multiple strokes, like writing a capital letter. Thus, this limits the environments where ultrasound sensing can be applied. Systems like UG [21] can recognize multi-stroke hand movements but require four transceiver pairs to work effectively. Since commercial off-the-shelf mobile devices typically have at most a speaker and two microphones, such approaches are hardly to deploy on commercial devices. Different from LLAP [20] and UG [21], which use fixed-frequency active sound waves for gesture recognition, WordRecorder [22] and Ipanel [24] use a microphone to record the sound generated by the friction between fingers and the table for gesture recognition. The aforementioned two methods, though, can recognize multi-stroke gestures with just one microphone. On the one hand, however, their performance is limited to adjacent 2D surfaces, for example, surfaces without sound (e.g., cotton surfaces) or 3D air hardly work. On the other hand, their performance is sensitive to environmental noise and can only operate in a quiet environment.

We present UltraScr, a novel sensing system based on ultrasound waves. UltraScr is designed to capture 3D in-air gestures (26 capital letters, 10 numbers, and 12 common interactive gestures) using a single transceiver pair. Unlike other systems [22, 24], UltraScr is not affected by background noise due to high-frequency sound waves, and the user is not limited by time and location in drawing the 3D in-air gestures. Note that UltraScr's signal sensing is independent of surface material, which indicates that it can expand the interaction area of mobile devices with bounded screens (e.g., smart wristbands). Moreover, by evaluating UltraScr in different users, different operations, and various locations, it can achieve accuracy up to 98% but only needs a single sound transceiver pair. Also, to avoid model invalid caused by changed conditions, UltraScr uses the rejection classification method (RCM) and incremental learning to update the training dataset and retrain the model.

Specifically, UltraScr uses the frequency attenuation profile (FAP) for gesture recognition. Although prior work like WordRecorder [22] and Ipanel [24] also use the sound frequency-domain information, essentially the same as FAP, for gesture recogni-

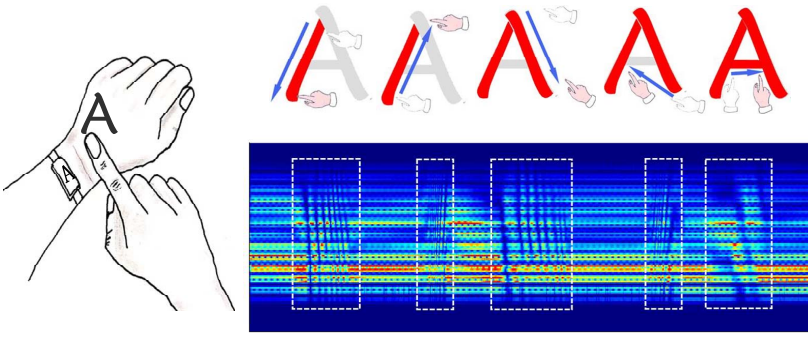


Figure 1: A typical use case of UltraScr. The built-in speaker and microphone of the smart bracelet emit and receive ultrasound waves, and when writing a capital letter 'A', different strokes have different FAP.

tion, the frequency-domain features used in their work are unstable as well as easy to be affected by the surrounding environment noise. Instead, UltraScr uses an active, stable and high-frequency ultrasound signals so that it overcomes the drawbacks of WordRecorder and Ipanel. Intuitively, as the hand moves, the reflected path of the ultrasound wave changes, so does the attenuation of the high-frequency ultrasound wave. This phenomenon implies that gesture points distance from the microphone could be measured by attenuation. It is easy to understand that once we extract the frequency attenuation of the ultrasound band at every time point, these frequency attenuation could be combined to form a FAP, in other words, FAP seems like an image of the gesture. So we could use the FAP to represent one gesture. Figure 1 illustrates a typical use case of UltraScr. Here, the built-in speaker of the smart bracelet transmits the modulated high-frequency sound waves in real-time, with a specific frequency band ranging from $17kHz$ to $21kHz$ [29]. Meanwhile, the built-in microphone of the wristband records the echo signals of the transmitted signal in real-time. During writing the capital letter 'A', the frequency attenuation change of echo signal recorded by the microphone is analyzed in real-time. Different strokes of writing 'A' have different FAP.

Inspired by the above analysis, we think it is feasible for applying the FAP extracted from the echo signal to recognize gestures. Following this intuition, we get the FAP of each gesture through using the Short-Time Fourier Transform (STFT). Then we turn the wireless gesture recognition into an image classification task. So a convolutional neural network (CNN) model is built to extract gesture features from the spectrum of the wireless signal. Finally the extracted gesture features are inputted into a learned support vector machine (SVM) classifier for predicting which gesture the user drew.

UltraScr has the following advantages over prior work:

- No additional equipment needed to be deployed. We use a speaker and a microphone that already exist on most mobile devices to perform device-free tracking of a hand/finger. Instead, UG [21] uses a custom speaker-microphone kit with 2 speakers and 4 microphones.

- It is not affected by other noises, such as speaking in the environment. Because the extracted features are high-frequency features, and the noise in the environment is

mostly low-frequency. Instead, the recognition accuracy of WordRecorder [22] and Ipanel [24] drops to about 50% when the ambient noise reaches 60dB.

- UltraScr allows users to write capital letters according to their habits, without forcing the user to write a letter in a single stroke. In contrast, LLAP [20] requires the user to write one letter at a time in a single stroke.

The contributions of our work are summarized as follows:

- We propose a novel 3D ultrasound sensing system UltraScr for gesture recognition to enable text input and gesture control of mobile devices. Note that the gestures in this article refer to 26 capital letters, 10 numbers, and 12 common interactive gestures. In particular, UltraScr can perform gesture recognition in three locations: in the air, on the desktop, and on the back of the hand.
- We propose the rejection classification method (RCM) that rejects wrong identification results and accepts correct identification results. RCM is a particularly critical technique to determine when incremental learning is needed to improve the system's robustness.
- UltraScr uses the FAP of high-frequency sound waves to represent hand movements, features are extracted from the CNN model, and gestures are recognized by the SVM model. Experimental results show that UltraScr can achieve up to 98% accuracy for trained users but only needs a single sound transceiver pair and 86% for users without training. Also, UltraScr requires 3× fewer training samples compared to the state-of-the-art.

2. Background

2.1. Ultrasound Sensing

UltraScr leverages the ultrasound wave for gesture recognition. It is achieved by using a sound transceiver pair, i.e., a speaker (transmitter) and a microphone (receiver). The speaker continually emits high-frequency sound waves, which propagates through the air and can bounce off the wall, furniture, and human body. By measuring how the sound wave is uniquely affected by a hand movement and comparing the measurement against pre-collected training data, UltraScr can infer what gesture has been performed.

2.2. Limitations of Current Approaches

In addition to not requiring gestures to be completed in a single stroke, UltraScr addresses two significant drawbacks in current ultrasound sensing solutions, the impact of microphone number, and the impact of sensing environments. To further illustrate these limitations, we consider following two case studies.

Impact of the microphone number. Prior work [21] has illustrated that the gesture recognition accuracy can be improved by increasing the number of microphones. On the one hand, electronic components will lead to high production costs; on the other hand, adding electronic components will increase the size of the device, and smartphones or other wearable devices typically come with up to two microphones. UG [21] is the state-of-the-art ultrasound sensing system for gesture recognition. As Figure 2a shows, UG

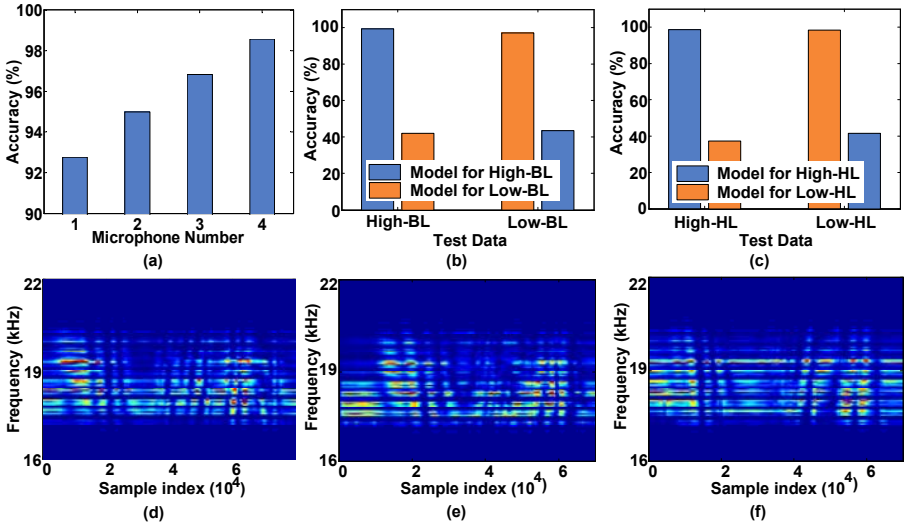


Figure 2: Accuracy and FAPs of prior work under different number of microphones and different external conditions. Specifically, (a), (b) and (c) respectively present the accuracy under the number of microphones, Battery Levels and Humidity Level. (d), (e) and (f) present the FAP of the capital letter 'X' under different external conditions. (d) shows the FAPs in High-BL and Low-HL, (e) shows the in High-BL and High-HL, and(f) shows the FAP in Low-BL and High-HL. They indicate that the features of spectrum images are different under different BL and HL.

can only achieve 92% by using one pair of speaker and microphone, which is 6% lower than the best accuracy achieved through four pairs of speakers and microphones.

Impact of battery and humidity. In this experiment, we recruited four volunteers to conduct experiments at different Battery Levels (BL), i.e., 90% BL (High-BL) and 10% BL (Low-BL), and different ambient Humidity Levels (HL), i.e., 90% HL (High-HL) and 60% HL (Low-HL). Detailed experimental settings are shown in Section 6.5. Figure 2d, 2e, and 2f show that under different external conditions (i.e., different BL and different HL), the FAP of echo signal collected when the same volunteer writes the same capital letter (i.e., 'X') is visually different. We respectively calculate the recognition accuracy of the model under both the same and different external conditions. Specifically, under the same external conditions, we calculate the average recognition accuracy through 10-fold cross validation; under different external conditions, training and testing data are collected in different external conditions. For example, if training data is collected in High-BL condition, test data need to be collected from Low-BL condition. Figures 2b and 2c show that the model's recognition accuracy is significantly reduced when the model's training and test data are collected at different BL and HL, respectively. Note that what we are showing here is the result of recognizing 26 capital letters. In particular, as shown in Figure 2b, when Model for Low-BL is used to test the data collected under High-BL, the recognition accuracy drops from 98% to 41%. The reason behind this is that the attenuation of sound waves is not consistent in different propagation media (i.e., different HL), and the energy of ultrasound transmitted by mobile devices is also different in different BL [30]. We will show the impact of changes in external conditions on UG in Section 6.5.

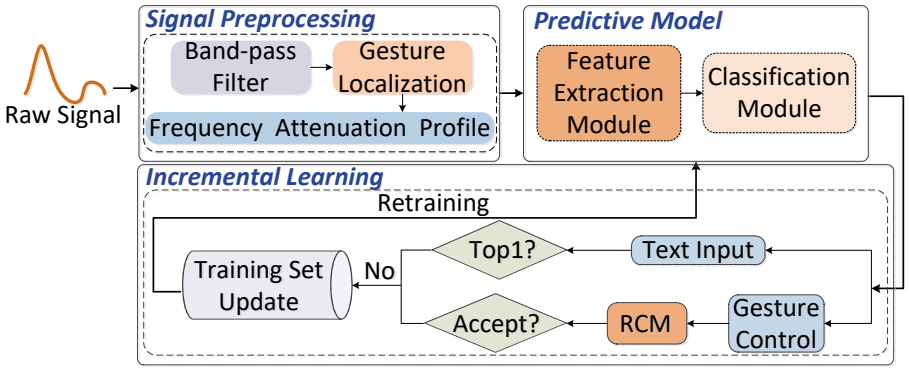


Figure 3: Overview of UltraScr. First, the raw signal recorded by the microphone is preprocessed. Then the features are extracted, and the gestures are recognized through the predictive model. Also, according to different applications, different methods are proposed to determine whether incremental learning is needed to improve the robustness of UltraScr.

Lessons learned. These examples illustrate the need to implement an ultrasound sensing system on mobile devices using a single sound transceiver pair, and the system should be robust to changes in external conditions.

3. Overview of UltraScr

Figure 3 gives an overview of UltraScr. The built-in speaker of the mobile device sends out a modulated high-frequency sound wave, whose echo signal is captured and recorded by the built-in microphone. The echo signal is then preprocessed to extract the signal characteristics that can represent different gesture operations. Next, the extracted signal characteristics are used to recognize the gesture through the predictive model. To adapt to changes in external condition, UltraScr employs incremental learning and rejection classification method (RCM) to retrain the predictive model. This is because variations in external conditions such as humidity level and battery level can interfere with ultrasound propagation (Section 2.2), which causes severe degradation in the sensing system's accuracy. To solve this problem, RCM defines four statistic and assessment features (see Section 5.1) to determine if a given sample's predicted result is correct.

3.1. Signal preprocessing

Many other frequency voice signals will be recorded in the echo signal, so we first filter the echo signal with a band-pass filter, then find the start and end positions of the gesture with gesture localization algorithm widely used by prior work [21] in wireless sensing (see Section 4.2 for details). Finally we get the FAP of each gesture through spectrum conversion.

3.2. Predictive model

UltraScr consists of a CNN model and a classic SVM classifier. The CNN model is used to extract gesture features, and the SVM classifier is applied to make a final prediction. Specifically, we feed the gesture features extracted by the CNN model to the

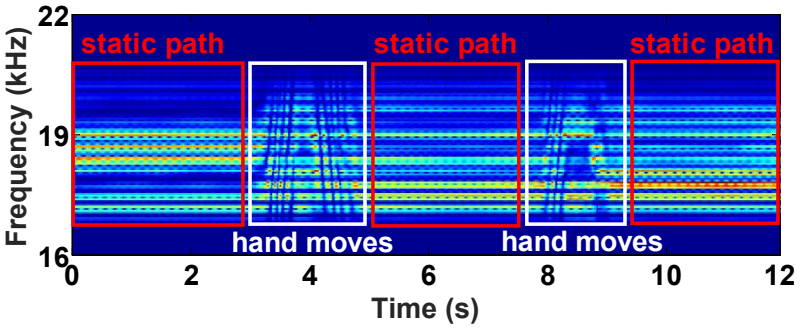


Figure 4: The static path and dynamic path.

SVM classifier, which then outputs the predicted labels. The CNN model is comprised of five convolutional layers, three pooling layers followed by three fully-connected layers. Likewise other neural networks, we also use dropout to avoid model overfitting. The specific model structure and hyperparameters setting are shown in Section 4.3.

3.3. Incremental learning

To improve the robustness of the system, we propose a rejection classification method (RCM) with the ability of rejection for wrong results and classification for correct results. When RCM gives a hint to reject classification, we update the training set and retrain the model. Note that RCM is a particularly valuable technique for determining when incremental learning begins (see Section 5).

4. Predictive modeling

Building and using a predictive model follows the well-known 4-step process for supervised learning: (1) modeling the problem domain, (2) generating training data, (3) learning a predictive model, and (4) using the predictor. These steps are described as follows.

4.1. Modeling the Problem Domain

The propagation of the sound wave is a mechanical phenomenon, which produces a mechanical wave that transmits initial energy through a medium. When the sound wave propagates in the air, the propagation path is mainly divided into the static path and the dynamic path. The static path corresponds to the sound wave traveling through the Line-of-Sight (LOS) path or being reflected by static objects, such as tables and walls. The state of the channel transmitted by the static path does not change; that is, the transmission function of the channel does not change. The dynamic path refers to the reflection caused by a moving object, such as a moving hand when making a gesture. When the hand moves, the channel reflection path will change, so that the transmission function of the channel will change. Figure 4 shows the frequency energy changes under the static path and dynamic path, respectively. The frequency energy of the static path hardly changes with time. However, the energy of different frequencies of the static path

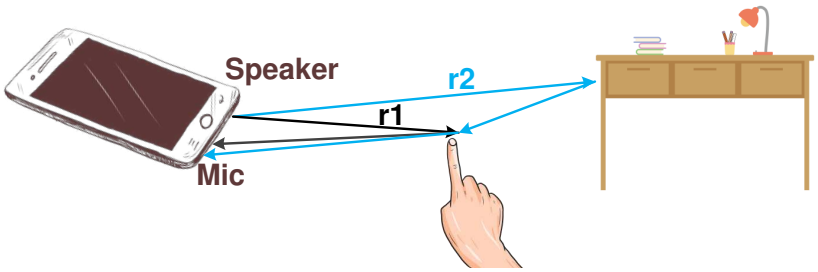


Figure 5: A simple example of the multipath effect.

varies due to the wireless signal multi-path characteristics, as shown in the red rectangle in Figure 4, which is also called frequency selective fading. When the hand moves, the arrival times of the multi-path signals are different, causing the energy of the same frequency to change significantly over time, as shown in the white rectangle in Figure 4, which is also called time selective channel.

Furthermore, under the multipath effect, the time of signal propagation varies along different paths. Therefore, the sound wave received by the microphone is composed of multiple delay signals, which can be expressed by Equation 1,

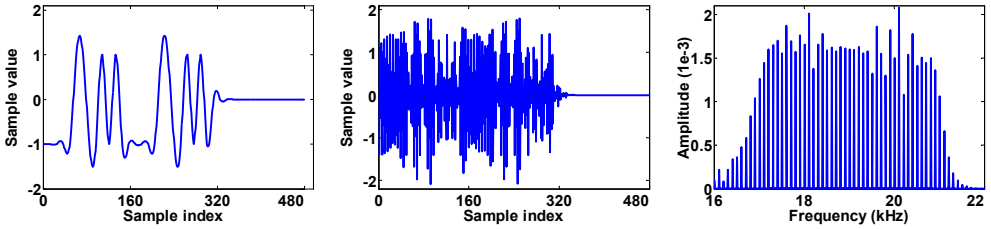
$$h(t) = \sum_{i=1}^{N} a_i \delta(t - t_i) + s(t) \tag{1}$$

where a_i represents the attenuation coefficient of the path i, $\delta(t - t_i)$ represents the signal received from the path i at time t, and $s(t)$ represents the signal received from the static multipath at time t. Specifically, we take a simple example with two multipaths to illustrate the signal received under the multipath effect. Note that in practice, such a situation is almost non-existent; we are here only to illustrate the impact of multipath. As shown in Figure 5, the speaker emits a cosine ultrasonic wave of frequency f, denoted by $cos\,(2\pi ft)$, $r1$ and $r2$ represent two multipath, whose transmission distance is d_1 and d_2, respectively, and the transmission speed of sound waves is denoted by c. Therefore, the signal received by the microphone can be expressed as:

$$\begin{aligned} h\,(t) &= a_{r1} cos\left(2\pi f\left(t - \frac{d_1}{c}\right)\right) + a_{r2} cos\left(2\pi f\left(t - \frac{d_2}{c}\right)\right) \\ h\,(s) &= a_{r1} cos\left(2\pi f\left(\frac{s}{f_s} - \frac{d_1}{c}\right)\right) + a_{r2} cos\left(2\pi f\left(\frac{s}{f_s} - \frac{d_2}{c}\right)\right) \end{aligned} \tag{2}$$

Where, a_{r1} and a_{r2} represent the attenuation coefficients of path $r1$ and $r2$, respectively. Note that the relationship between time t and the sample index s is $t = \frac{s}{f_s}$, where f_s represents the sampling rate. Here, path $r2$ represents the static path, and the signal from path $r2$ corresponds to $s(t)$ in Equation 1.

The synthesized signal received by the microphone will show time delay expansion relative to the original signal in the time domain. The size of the delay expansion can be intuitively interpreted as the difference between the maximum propagation delay and the minimum propagation delay, i.e., the difference between the arrival time of the last



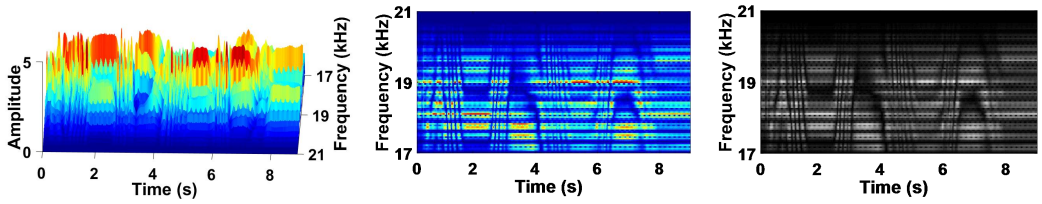
(a) Sequence. (b) Modulated signal. (c) Frequency.

Figure 6: Modulated signal and its frequency.

distinguishable delay signal and the arrival time of the first delay signal. From the perspective of the frequency domain, delay expansion will lead to frequency selective fading; that is, the attenuation of each frequency component is different for different channels and signals. The coherence bandwidth is an important indicator to describe the delay expansion, which is approximately equal to the reciprocal of the maximum delay, i.e., $B_c = \frac{1}{\Delta t} \approx \frac{1}{t_{max}}$. When the channel bandwidth is larger than the coherence bandwidth, the change of each frequency component after the signal is transmitted through the channel is inconsistent, which will cause waveform distortion, and the fading is called frequency selective fading. Conversely, when the channel bandwidth is smaller than the coherence bandwidth, the changes of each frequency component after the signal is transmitted through the channel are consistent, the signal waveform is not distorted. Thus the fading is flat fading (non-frequency selective fading). Back to the UltraScr, when the target is less than $0.25m$ away from the microphone, the maximum reflected distance of the signal is $0.5m$, the maximum propagation delay $t_{max} = \frac{0.5m}{340m/s}$. Since the speaker and microphone on the mobile device are always close together, the minimum propagation delay is approximately 0. In this case, the coherent bandwidth $B_c = \frac{1}{\Delta t} \approx 680$. Thus, frequency selective fading occurs when the bandwidth of the transmitted signal is higher than $680Hz$. Therefore, in this paper, we propose gesture recognition based on the FAP. The more high-frequency sound waves are sent, the more granular the FAP is, and the more unique the gesture features can be. In order to make the high-frequency component of the transmission signal as much as possible, we use sequence modulation to modulate the transmission signal higher than $17kHz$.

Signal Modulation. In this paper, the 13-bit Barker code is extended to 480 bits by copying, up-interpolating, and padding zeros to construct the sequence S (shown in Figure 6a) in the transmitted signal [21]. The carrier frequency we used is $f_c = 19kHz$. The modulated periodic signal can be expressed as $Signal = S \times cos(2\pi f_c t)$, where $t = 0, \frac{1}{f_s}, \frac{2}{f_s}, \cdots$, and f_s is the sampling rate. Figure 6b shows the modulated periodic signal $Signal$. In order to get complete gesture feedback, the speaker repeatedly transmits the modulated periodic signal. The frequency of the transmitted signal is shown in Figure 6c, which is rich enough for gesture recognition. In this paper, we mainly use frequency information from $17kHz$ to $21kHz$.

Frequency Attenuation Profile (FAP). We use features extracted from FAP for gesture recognition, which outperforms other works [20, 21, 31, 32]. As the hand moves, FAP acquires variations in frequency attenuation, the position of the hand at each time



(a) Three-dimensional image. (b) Two-dimensional color image. (c) Two-dimensional gray image.

Figure 7: Explain the generation process of FAP. (a) shows the change of frequency attenuation at each time point during the writing of the capital letter A. The size of the sliding window is $200ms$. The changes in frequency decay at all points in time together form the FAP for capital letter A. (b) and (c) show two-dimensional color FAP and two-dimensional gray FAP, respectively.

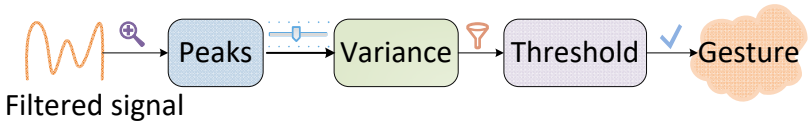


Figure 8: Overview of the gesture localization approach.

point can be represented by a frequency attenuation diagram. Figure 7a shows the frequency attenuation diagram of each time point when the capital letter 'A' is written. In order to facilitate processing and reduce the amount of computation, we compress FAP into two-dimensional and gray-scale processing, which are shown in Figure 7b and 7c, respectively. There are three strokes in the capital 'A', so there are three frequency attenuation fluctuations. The first wave represents slash '/' on the left side of 'A', the second wave represents slash '\' on the right side of 'A', and the third wave represents horizontal bar '-' in the middle of 'A'.

4.2. Generate Training Data.

After transmitting the high-frequency sound wave through the speaker of mobile devices and collecting the reflected signal from the microphone of mobile devices, we need to extract the FAP of the gesture from the collected sound waves. Firstly, the collected sound waves need to be preprocessed, including filtering out other frequencies, gesture localization, and FAP generation.

Band-pass Filter. Since the microphone will record sound waves in many other frequency bands, We use a Butterworth band-pass filter to leave only $17kHz$ to $21kHz$ sound waves.

Gesture Localization. We found that in the time domain, hand movement also affects the amplitude of the signal. Therefore, in order to locate the gesture, we first look for the peak value of the filtered signal, with a peak interval of at least 460 sample points (line 3 in Algorithm 1). A sliding window is then used to calculate the variance of the peak, and we set the sliding window size to 10 sample points. According to the threshold of peak variance set empirically, we obtain the dynamic path in the sound wave (line 4-9 in Algorithm 1). Finally, we use the time threshold to determine the start and end

Algorithm 1: Gesture Localization Algorithm

Input : The filtered sound wave S and its sampling rate fs; the sliding window size W_s; the minimum peak interval P_i; variance threshold V_t; time threshold T_s.

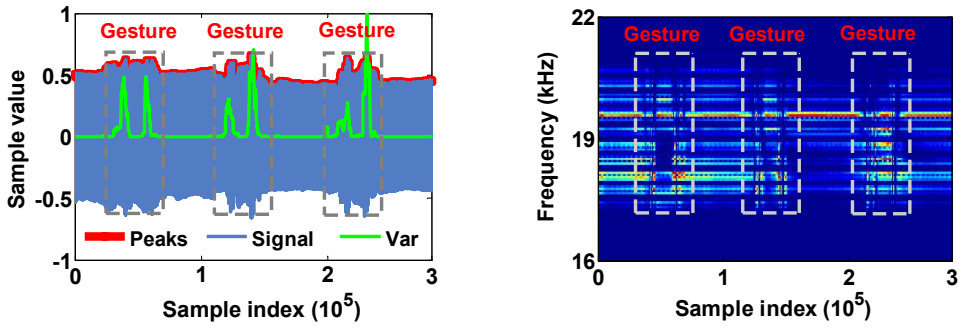
Output: The set of dynamic path W; the set of gesture start point $\mho_s$; the set of gesture end point $\mho_e$.

1 Initialize: $W = \Phi$; $\mho_s = \Phi$; $\mho_e = \Phi$; $t = 1$.
2 /*Locating all the peaks and their locations in the filtered sound wave.*/
3 $[pks, locs] = findpeaks(S, W_s)$;
4 **for** *each peak* $p(i) \in pks$ **do**
5 $v(i) = Var(p(i : i + W_s))$;
6 **if** $v(i) \geq V_t$ **then**
7 $W(t) = i$; $t = t + 1$;
8 **end**
9 **end**
10 **for** *each dynamic path* $t \in W$ **do**
11 **if** $locs(t) - locs(t-1) > T_s \times fs$ **then**
12 $\mho_s \leftarrow locs(t)$;
13 **end**
14 **if** $locs(t+1) - locs(t) > T_s \times fs$ **then**
15 $\mho_e \leftarrow locs(t)$;
16 **end**
17 **end**

of the gesture (line 10-17 in Algorithm 1). The process of gesture localization is shown in Figure 8. Algorithm 1 shows the implementation process of the gesture localization algorithm. In UltraScr, $fs = 48kHz$, $W_s = 10$, $P_i = 460$, $V_t = 0.2$. Also, some gestures and capital letters & numbers cannot be written in one stroke, and there is an extremely short pause between different strokes in the same letter. Therefore, we introduce a time threshold T_s to distinguish different operations and different parts of the same operation. Through benchmark experiments, we found that, under normal circumstances, the pause in the same operation does not exceed 0.6 seconds, and the interval between different operations must be greater than 0.6 seconds, thus $T_s = 0.6s$ in our system. After obtaining the start point and endpoint of the gesture in the time domain, we intercept the corresponding gesture segment on the FAP as the input of the feature extraction model.

Figure 9 shows the gesture segmentation of the gesture localization algorithm when three gestures are made in succession. The result of gesture segmentation is one-to-one correspondence in the time domain and the frequency domain.

Frequency Attenuation Profile. As explained in Section 4.1, gesture recognition is mainly based on time-frequency domain features. So we use the STFT to obtain the time-frequency features of the high-frequency part. We choose the window size of STFT to be 1024, the overlap size to be 512 and do Fast Fourier Transform (FFT) every 1024 points. In order to reduce the amount of computation, we intercept the part of the



(a) Time domain. (b) FAP.

Figure 9: Results of gesture localization.

spectrum from $17kHz$ to $21kHz$ and gray-scale it. Thus, we get the FAP for each gesture.

4.3. Learning a Predictive Model

Our predictive model consists of a CNN feature extraction module and an SVM classification module. The feature extraction module is used to extract the features given a FAP of the ultrasound signal. The classification module aims to make the final prediction. The output of the feature extraction module is taken as the input of the classification module. It's well know that building a robust CNN model requires thousands of training samples, otherwise, the model is particularly prone to overfitting if lack of enough training samples. However, there is impractical and hard to get so many gesture data for training. In addition, it is challenging to train CNN on mobile devices with limited computing resources due to its computational complexity. Combining the aforementioned considerations, here we only use CNN to extract the signal features instead of for classification due to lack of enough training samples. Furthermore, the effect of the SVM classifier is better than the CNN classifier when there is a small number of training samples. Thus, we can train the CNN feature extraction module offline and save it to the device, and when the model needs to be updated, we only update the SVM classification module. Zhou et al. [33] shows that one of the advantages of using CNN-SVM is that the CNN model can be trained offline to extract features and save the model. Only the SVM needs to be retrained on the mobile device when a new classification model needs to be trained. In particular, in our study, we update the model through incremental learning to ensure its stability.

Feature Extraction Module. UltraScr uses the CNN model to extract gesture features, which reduces the feature dimension and extracts more distinguishing features. CNN has shown particularly remarkable performance in image processing and recognition [34, 35]. First of all, we use some classic CNN network structures, such as ZFNet [36], LeNet [37], VGG-13 [38], and AlexNet [39], to adjust the parameters to fit our problem. Then, after preliminary analysis and testing, we found that the structure of AlexNet and VGG-13 is too complicated and needs too many parameters, which would affect the training time of the model and the response time of the results. Therefore, we consider

Layer	Layer Type	Filters/Units	Size	Strides	Activation
1	Conv2D	32	8×8	[2,2]	tanh
2	MaxPooling2D	-	5×5	[2,2]	-
3	Conv2D	64	8×8	[1,1]	tanh
4	MaxPooling2D	-	5×5	[2,2]	-
5	Dropout	-	40%	-	-
6	Conv2D	64	5×5	[1,1]	tanh
7	Conv2D	64	5×5	[1,1]	tanh
8	Conv2D	32	5×5	[1,1]	tanh
9	MaxPooling2D	-	4×4	[2,2]	-
10	Dropout	-	40%	-	-
11	Flatten	-	-	-	-
12	Dense	256	-	-	tanh
13	Dropout	-	75%	-	-
14	Dense	128	-	-	tanh
15	Dropout	-	75%	-	-
16	Dense	class number	-	-	softmax

Table 1: CNN layers and hyperparameters setting.

combining ZFNet and LeNet to construct the CNN network structure. By continually adjusting the structure and parameters, the CNN network structure is composed of five convolution layers, three pooling layers, three fully connected layers, and one output layer. To extract the multi-angle features, the size of convolutional kernels are 8×8, 8×8, 5×5, 5×5, and 5×5 for each layer respectively, and the pooling size is 5×5, 5×5, and 4×4. The activation function used in the CNN network structure is $tanh$, except the last full connection layer is $softmax$. We also use the dropout layer to avoid model overfitting. Details of the parameters of the CNN network structure are shown in Table 1. As an example, Figure 10 shows the features of capital letters 'A' and 'B' extracted from the penultimate layer of the CNN network structure, resulting in a 128-dimensional feature for each gesture. The features of capital 'A' and 'B' are easily distinguishable. In particular, after the CNN model has been trained, the last fully connected layer of the CNN model is removed. In this way, the remaining network structure can be used as the feature extraction module of the system. Section 6.3 will show the classification results of features extracted from different layers of the feature extraction module.

Classification Module. Different classifiers have different classification effects on different features. We select eight different classifiers for experiments, including supervised and unsupervised classifier, linear and non-linear classifier, probability-based classifier, integrated classifier, and deep learning classifier. The eight classifiers are Logistics Regression (LR, penalty='l2', solver='sag', C=100), Support Vector Machine (SVM, C=1, kernel='linear', gamma=0.1), k-nearest-neighbor (KNN, n_neighbors=12), Decision Tree (DT, criterion='entropy'), Random Forest (RF, n_estimators=100, max_depth=20, ran-

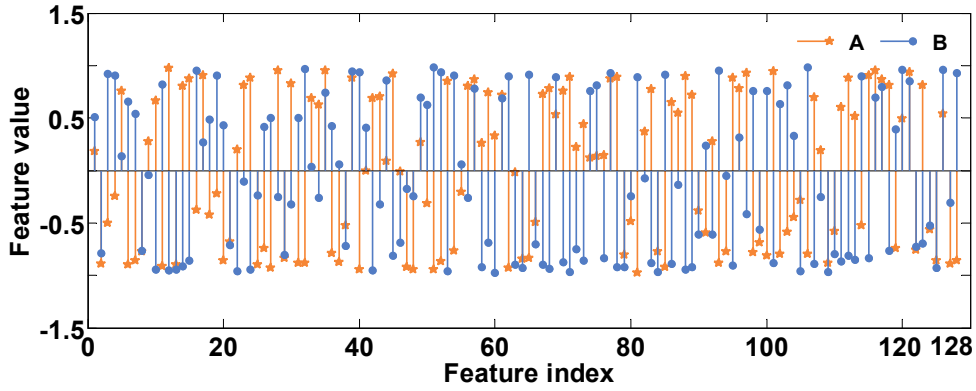


Figure 10: Features of capital letters A and B extracted by CNN.

dom_state=0), GaussionNB (GNB), Gradient Boosting (GB, n_estimators=10, learning_rate=0.01, max_depth=5, random_state=0), and CNN. As shown in Section 6.3, by comparing the results of different classifiers, we finally choose the SVM classifier. The parameters of these classifiers are set by grid search with 10-fold cross-validation.

4.4. Using the predictor

When the user uses the system, UltraScr uses the same data preprocessing method to process the echo signal recorded by the microphone. Then, the processed information is used as the input of the CNN feature extraction module to obtain more characteristic features. Finally, the output of the CNN feature extraction module is employed as the input of the SVM classifier to acquire the final recognition result.

5. Incremental learning

As shown in Section 2, the changes in external conditions, like battery level or humidity level, can have a significant impact on the prediction accuracy of the model. The reason behind this is that models trained in previous external conditions often make ambiguous and wrong decisions when faced with changing external conditions. This can be illustrated by Figure 11, which shows the predicted probabilities of a sample among all possible classes when the model is trained at a 60% humidity level and tested at 60% and 90% humidity level, respectively. We select 5 out of 26 capital letters to illustrate the ambiguity of model recognition, including 'D', 'O', 'P', 'U' and 'X'. The main reason we chose these five capital letters is that when external conditions change, they are difficult to distinguish due to the similarity of strokes, such as 'D' and 'P', 'O' and 'U'. Figure 11a shows that when the model is trained in the same external environment like the one being tested (i.e., 60% humidity level), the probability of the model identifying the sample as a true class is almost 1, and the probability of identifying the sample as other classes is almost 0. However, as shown in Figure 11b, when the external environment in which the model is tested is inconsistent with the external environment in which the model is trained (i.e., trained at 60% humidity level and tested at 90% humidity level), the recognition results of the model are ambiguous and even wrong. For example, the

	Prediction Probability				
	D	O	P	U	X
D	0.99	0	0.01	0	0
O	0	0.99	0	0.01	0
P	0.01	0	0.99	0	0
U	0	0.01	0	0.99	0
X	0	0	0.01	0	0.99

(a) 60% humidity level.

	Prediction Probability				
	D	O	P	U	X
D	0.48	0	0.52	0	0
O	0	0.23	0	0.77	0
P	0.53	0	0.47	0	0
U	0	0.71	0	0.29	0
X	0.01	0	0.47	0.06	0.46

(b) 90% humidity level.

Figure 11: The prediction probabilities of a sample among all possible classes. The model was trained at a humidity level of 60%, tested at a humidity level of 60% (a) and a humidity level of 90% (b).

probability of capital 'D' being identified as capital 'P' is similar to that of capital 'D'. Even capital 'P' has a slightly higher prediction probability than capital 'D'. The sample will be incorrectly identified as capital 'P' by the system. Therefore, in order to improve the robustness of the system, we need to retrain the classification model by incremental learning. However, when to begin to refine the trained model is the critical problem we main concern. If the model is frequently updated, it will decrease the availability of the system. On the contrary, the robustness of the system will decrease if the updating process dose not done in time. To solve this problem, UltraScr proposes an incremental learning framework based on online feedback, which newly uses a rejection classification method (RCM) to determine when to retrain the model correctly.

As mentioned in Section 1, UltraScr is a novel 3D ultrasound sensing system which can realize text input and gesture recognition. For text input, the recognition result is displayed on the screen, and the user clearly knows whether the result is correct or not [40]. If the identification result is incorrect, the user can use the sample and its label to further update the training data set. In particular, when the user starts writing letters or numbers, the recognition results of the top 3 are displayed on the screen of the mobile device, and the user selects the correct one. We chose the top 3 because we found the accuracy of the top 3 can reach about 90% when the environment changes. If the correct result is not top 1, then put the data and the right label into the updated training set. For gesture control, different from text input, incorrect recognition results will directly lead to incorrect control. For example, misidentifying the open network as setting the phone to airplane mode could result in missing important calls. Therefore, the method used for text input cannot be used for gesture control. For gesture control, it is necessary to output the correct classification results and reject the wrong classification results. In this paper, a rejection classification method (RCM) with the ability of rejection for wrong results and classification for correct results is proposed. For the classification results of the predictive model, RCM determines whether the results are accepted or not. If not, the sample and its label need to be used to update the training data set. In Section 6.5, Error Rejection Rate (ERR), the proportion of samples that are falsely rejected among

all samples, is used to evaluate the performance of RCM. The implementation details of the RCM are shown in Section 5.1.

5.1. Rejection Classification Method (RCM)

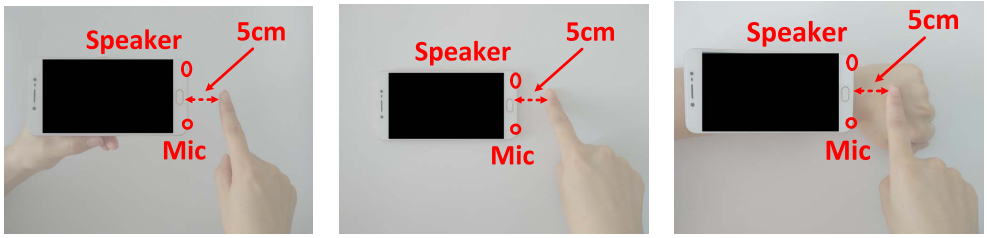
The purpose of the RCM is to reject incorrect classification results. One reason for the misclassification of predictive models is that the classification results of a test sample are based on other classes. In other words, a test sample is more likely to fall into one class than another. As shown in Figure 11b, changes in external conditions (e.g., changes in humidity level) may lead to more than one class matching of test samples. For example, when the true class of the test sample is capital 'D', the model predicts that the probability of the test sample being capital 'D' and 'P' is similar, which leads to misclassification. Therefore, RCM defines four features for each class according to the training data set: (1) Probability Value (PV), (2) Probability Confidence (PC), (3) Statistical Value (SV), and (4) Statistical Confidence (SC). If the classification results of a test sample by the predictive model do not meet the four features of the corresponding class, RCM considers the classification result incorrect and rejects the classification result. The four features are calculated as follows:

(1) Probability Value (PV). PV refers to the probability that the sample belongs to a class compared to other classes, which can be calculated according to the *predict_proba* [41] method in the *scikit_learn* [42] function. *Scikit_learn* is a widely used machine learning module that integrates a large number of state-of-the-art machine learning algorithms to solve supervised and unsupervised problems. *Predict_proba* calculates the probability that the sample is classified into each possible class based on the classification decision made on the sample by a classifier.

(2) Probability Confidence (PC). PC refers to how certain the probability that the sample falls into a class, which can be calculated according to $PC = 1 - max(PV_S \setminus PV_S^*)$, Where $PV_S = \left\{PV_S^1, PV_S^2, \cdots, PV_S^l\right\}$ represents the probability value of sample S for each class and $PV_S \setminus PV_S^*$ represents the probability value of all classes except the specific class $*$. In particular, class $*$ can be the class corresponding to the training sample or the class predicted by the predictive model for the test sample.

(3) Statistical Value (SV). SV refers to the probability that the sample belongs to a class compared to all of its other members. However, there is no method to calculate SV in existing machine learning. How to calculate SV may be a big challenge. Fortunately, SV can be calculated based on the sound statistical foundations of conformal predictions [43]. Firstly, the nonconformity of sample S and class $*$ is calculated according to Equation 3. The parameters in Equation 3 are the same as those in PC. Then SV can be calculated according to Equation 4, where $N_{\Re}^* = \left\{N_{\Re_1}^*, N_{\Re_2}^*, \cdots, N_{\Re_t}^*\right\}$ represents the nonconformity set of verification set $\Re$ of class $*$. $num\left|N_{\Re}^* \geqslant N_S^*\right|$ represents the number of $N_{\Re_i}^*$ greater than or equal to N_S^* in $N_{\Re}^*$. $num\left|N_{\Re}^*\right|$ represents the number of samples in the verification set $\Re$ of class $*$.

$$N_S^* = \frac{1 - \left\{PV_S^* - max\left(PV_S \setminus PV_S^*\right)\right\}}{2} \tag{3}$$



(a) In the air. (b) On the desktop. (c) On the back of hand.

Figure 12: Different locations.

$$SV = \frac{num\,|N^*_{\Re} \geqslant N^*_S|}{num\,|N^*_{\Re}|} \tag{4}$$

(4) Statistical Confidence (SC). SC refers to how certain the probability that the sample falls into a class compared to all of its other members, which can be calculated according to $SV = 1 - max(SV^*_S \setminus SV_S)$, Where $SV_S = \left\{SV^1_S, SV^2_S, \cdots, SV^l_S\right\}$ represents the statistical value of sample S for each class and $SV^*_S \setminus SV_S$ represents the statistical value of all classes except the specific class $*$.

6. Evaluation

We recruited 14 volunteers (5 females and 9 males) from our university aged between 18 to 28 to evaluate the performance of UltraScr. Our IRB approved all the experiments, and the privacy of all volunteers was protected. In this section, we mainly evaluate the performance of UltraScr from the following three aspects:

1. **Does UltraScr work for different people?** We evaluate the availability of UltraScr for each of the 14 volunteers using 10-fold cross validation. Each of the 14 volunteers wrote 26 capital letters on the desktop in a comfortable manner they like, and each letter was repeated 30 times. Different from the one shown in Figure 12b, we did not limit the write position of the volunteers. After observation, 14 volunteers completed the letters approximately 3cm-15cm away from the mobile device's speaker and microphone. The purpose of this experiment is to prove the applicability of UltraScr to different people.

2. **Can UltraScr apply to different scenarios?** Multiple factors have been taken into account, including different operations (26 capital letters, 10 numbers, and 12 gestures), different locations (on the desktop, in the air and on the back of the hand), different distances (5cm and 20cm), different volume levels (50% and 100%) and different algorithm parameters, which we believe could affect the performance of the system. Different operations can extend the scope of use of UltraScr, such as gesture recognition can be applied to the gesture control of smart devices. Different locations are helpful for UltraScr to be deployed on different devices.

For example, completing the operation in the air is convenient for the system deployed in the smart home, and completing the operation on the back of the hand can be applied to the smart wristband. Different distances and volume levels are closely related to the energy of sound waves sent and received, so we believe they may affect the recognition accuracy of UltraScr. As discussed in Section 4.3, parameters of the algorithm, such as the output of which layer of the convolutional neural network (CNN) model is the feature and which classification algorithm is selected for the final classification, are also closely related to the performance of the system.

3. **Can incremental learning improve the robustness of UltraScr?** We evaluate incremental learning performance through experiments on different humidity levels and battery levels for mobile devices. In particular, in this experiment, the training set and the test set come from different humidity levels or battery levels. For example, we train the model with data collected at a 60% humidity level and test the trained model with data collected at a 90% humidity level. The reason behind this experiment is that, on one hand, sound waves are particularly susceptible to the influence of transmission medium, physical environment, etc. [30]; on the other hand, traditional gesture recognition systems based on machine learning are usually one-shot learning, so that when the transmission medium or physical environment changes, the learned model will become vulnerable. Therefore, the purpose of the experiment is to evaluate whether the rejection classification method proposed in this paper can correctly determine the time point of incremental learning and whether incremental learning can improve the system's robustness.

6.1. Experimental Setting

Hardware and Software Platforms. We implement UltraScr on a mobile phone (vivo Y67A, 1.5GHz eight cores, a RAM with 4GB, Android 6.0), which has one speaker and two microphones. UltraScr only uses a single sound transceiver pair, that is, the mobile phone's speaker and the microphone at the bottom. Specifically, like existing work [20, 44], we use mobile phones to simulate wearable devices (e.g., smart wristbands). We learned that most wearable devices are developed based on the Android system, so we believe that the application of UltraScr will not be affected when conducting experiments with mobile phones. The mobile phone supports a sampling frequency of $48kHz$ and can transmit and receive high-frequency sound waves over $17kHz$. Currently, some wearable devices cannot receive high-frequency sound waves temporarily due to the design of electronic components, so we use mobile phones to simulate wearable devices to evaluate the performance of UltraScr on them.

Implementation and Evaluation Platform. The prototype system is implemented in Java and Python programming languages. Specifically, the sending, receiving, and preprocessing of sound waves are realized as an APP on mobile devices by using Java programming language. The CNN extraction feature module and classification module are implemented in a multi-core cloud server (14-core 2.4GHz Intel Xeon CPU and an NVIDIA P40 GPU) using the Python programming language. In the real-time recognition stage, the mobile device forwards the processed data to the cloud server for feature extraction and recognition. Note that both feature extraction models and classification models are trained offline.

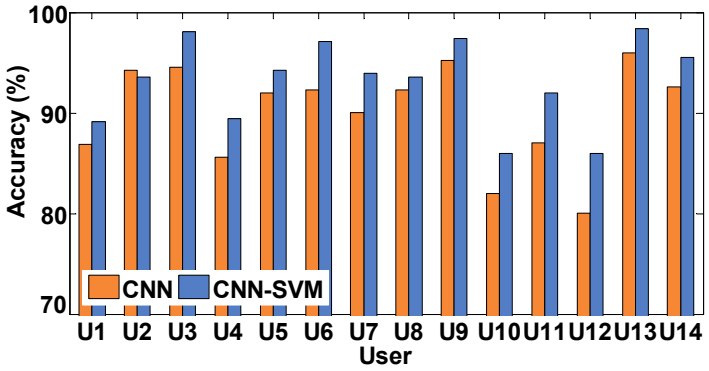


Figure 13: Accuracy among different users.

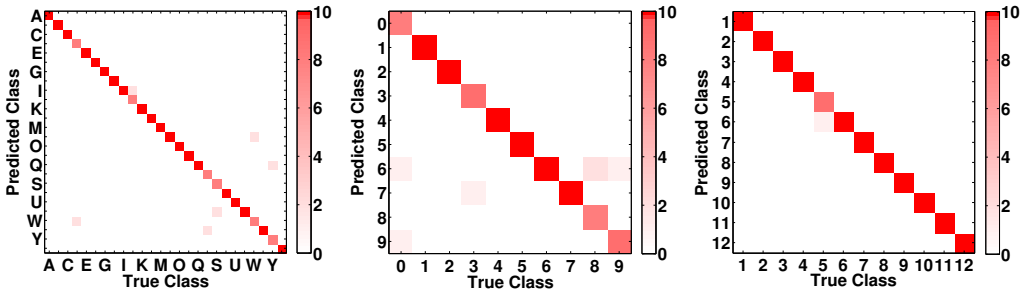
Data Set. By default, each volunteer writes 26 capital letters 30 times on the desktop according to their writing habits. A total of 10,920 (26 capital letters × 30 instances × 14 users) samples are collected. In addition, other data are collected from some of the volunteers, including two other operations ((10 numbers + 12 gestures) × 30 instances × 3 users = 1980 samples), three different locations (3 locations × 26 capital letters × 30 instances × 3 users = 7020 samples), two different volume levels (2 volume levels × 26 capital letters × 30 instances × 3 users = 4680 samples), two different distances (2 distances × 26 capital letters × 30 instances × 3 users = 4680 samples), two different humidity levels (2 humidity levels × (26 capital letters + 10 numbers + 12 gestures) × 100 instances × 2 users = 19200 samples) and two different battery levels (2 battery levels × (26 capital letters + 10 numbers + 12 gestures) × 100 instances × 2 users = 19200 samples). Detailed data collection will be covered in the following sections.

Experimental Environment. By default, each volunteer performs the experiment in the office where they work. More than 30 employees in the office carry out daily work, while air conditioning, fresh air system, and other equipment have been running. The noise in the office is measured at about 40dB-70dB by using the noise decibel meter APP on another mobile phone. The temperature and humidity of the office are about 26°C and 60%, respectively, measured by a universal thermo-hygrometer. For different humidity levels, the volunteers are asked to complete the experiment in a dedicated office, with the humidity in the environment controlled by a humidifier and air conditioner.

Note that during the whole experiment, we did not apply restrictions on the size, speed, stroke order, and other characteristics related to personal habits.

6.2. Accuracy Achieved by Different Users.

In this experiment, each volunteer (denoted as U1 to U14) is asked to place the mobile phone on their office table, open the APP, and write 26 capital letters on the desktop, repeating each one 30 times. Unlike Figure 12b, we do not limit the distance between the write position and the phone. The arrangement of items (e.g., computers, cups, bookshelves, etc.) on each volunteer's desk is different; therefore, the propagation of sound waves suffers from different multipaths. Figure 13 shows that most volunteers' recognition accuracy exceeds 90%, and the highest recognition accuracy can reach 98%.



(a) 26 capital letters. (b) 10 numbers. (c) 12 gestures.

Figure 14: Confusion matrix of different operations.

ID	Gesture	ID	Gesture
1	Flick	7	Hover-Left
2	Anti-Flick	8	Hover-Right
3	Clockwise	9	Click
4	Anti-Clockwise	10	Double-Click
5	Push-Pull	11	Swipe-Left
6	Pull-Push	12	Swipe-Right

Table 2: Gestures used in our evaluation.

In addition, Figure 13 also shows that the recognition accuracy of CNN-SVM is higher than that of CNN. At the same time, we observe slight variations in the accuracy of the different volunteers. The main reason behind this is that different volunteers have different writing habits, and we do not stipulate the users' stroke order. Some users write certain letters that are even hard for the human eye to tell apart, such as 'D' and 'P', 'O' and 'Q'. This problem can be addressed in future work, either by extracting more granular features or regulating the user's writing behavior.

6.3. Accuracy in Different Scenarios.

Different Operations. We first evaluate the performance of UltraScr in various operations. In this experiment, we invite three volunteers who are asked to complete 10 numbers and 12 gestures for 30 times, respectively, on the desktop. The 12 gestures we used are shown in Table 2. We divide the collected data set into a training set and test set at a ratio of 2:1. Thus, there are 20 training samples and 10 test samples for each class. Figure 14 shows the confusion matrix of 26 capital letters, 10 numbers, and 12 gestures identified by UltraScr. In particular, we calculate the average of three volunteers. Obviously, in all three operations, the system has strong identification capabilities.

Different Distances. Next, we evaluate the effect of writing position on system perfor-

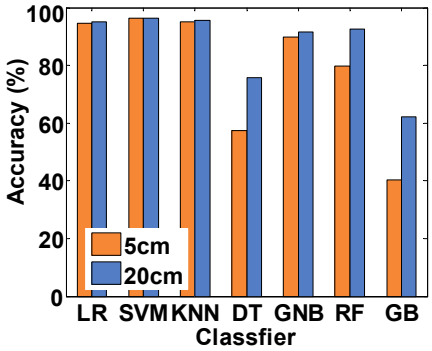


Figure 15: Accuracy of different distance.

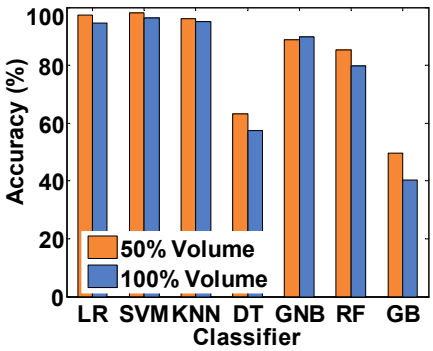


Figure 16: Accuracy of different volume.

mance. In this experiment, we invite three volunteers to participate and finally calculate the average accuracy of the three volunteers' 10-fold cross validation. Each volunteer is asked to complete the experiment at a distance of 5cm and 20cm from the mobile device, respectively. In each experiment, the volunteer is asked to write 26 capital letters 30 times on the desktop. Figure 15 shows that the recognition accuracy of UltraScr is similar when the writing position is 5cm and 20cm away from the mobile device. Therefore, we can infer that when the writing position is within 20cm from the mobile device, the performance of the system is basically unaffected by distance.

Different Volume Levels. Additionally, we evaluate the impact of the volume level of the mobile device on UltraScr. As shown in Figure 6c, the frequency range of our modulated sound wave is approximately $17kHz$ to $21kHz$. Although it is difficult for adults to hear more than $17kHz$ as they get older, some people and children can still hear these high-frequency sound waves. One way to reduce this effect is to reduce the volume level of the mobile device. Therefore, we invite three volunteers to participate in this experiment and finally calculate the average accuracy of the 10-fold cross validation of the three volunteers. Each volunteer performs experiments at 50% and 100% volume levels of the mobile device, respectively. In each experiment, we let each volunteer write 26 capital letters 30 times on the desktop. As shown in Figure 16, UltraScr achieves over 97% accuracy at both 50% and 100% volume levels. In particular, the system's accuracy is slightly higher at 50% volume level than at 100%. The reason behind this is that a sound wave at a 100% volume level has more energy than a sound wave at a 50% volume level, and therefore has more sophisticated multipath interference. Furthermore, it also means that high volume levels are not necessarily beneficial to our system.

Different Locations and Surface Materials. Since mobile devices are often carried around and used in different locations, we evaluate the performance of UltraScr when the user writes 26 capital letters in three different locations as shown in Figure 12, including in the air (location 1), on the desktop (location 2), and on the back of the hand (location 3). For the convenience of comparison, in this experiment, we require the writing position to be 5cm away from the mobile device. We invite three volunteers, and each volunteer repeats each capital letter 30 times in three different positions, respectively. Finally, the average accuracy of 10-fold cross validation of three volunteers at three different locations is calculated, respectively. As shown in Figure 17, when users write 26

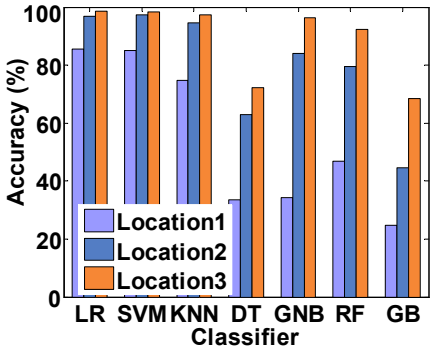


Figure 17: Accuracy when operating at different locations.

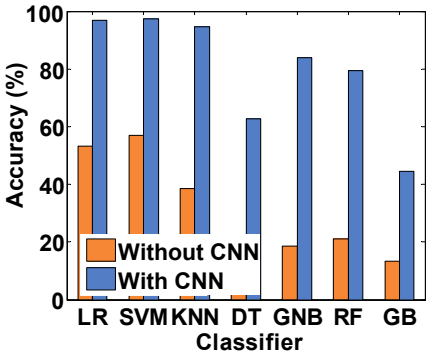


Figure 18: Accuracy of different classifiers with and without CNN.

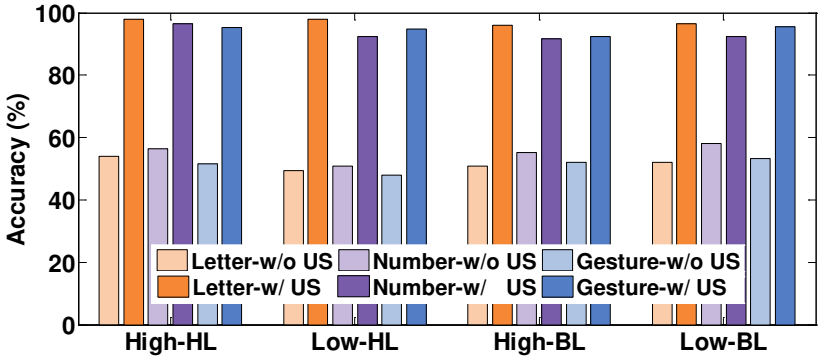
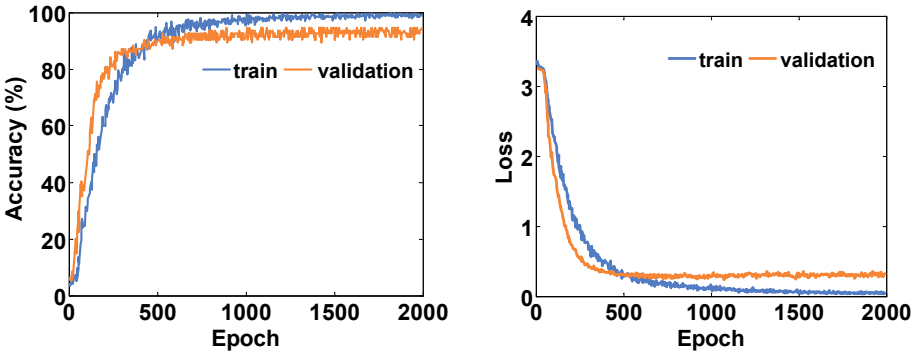


Figure 19: Accuracy of the models trained by using the data collected in multiple environmental conditions. The horizon coordinate represents the humidity and battery levels, e.g., High-HL, Low-HL, High-BL and Low-BL. For example, High-HL represents the model is tested using the data collected in high humidity level environment and trained using the data collected in other three external conditions. To comparison, we also calculated the recognition accuracy of our method, respectively.

capital letters on the desktop or on the back of the hand, the recognition accuracy of UltraScr exceeds 97%. Thus, unlike WordRecorder [22] and Ipanel [24], which recognize capital letters by the sound of a microphone recording pen/hand rubbing against a surface, the performance of UltraScr is independent of the surface material. The accuracy is slightly lower when writing 26 capital letters in the air, mainly because the finger movement in the air is less obvious when writing capital letters of the same size, and it is difficult to distinguish certain strokes of similar letters, such as 'D' and 'P'. Therefore, a slight increase in movement range would be helpful when using the system for gesture recognition in the air.

Multiple training datasets collected from different environmental conditions. In this evaluation, we wonder if using multiple training datasets collected in different external conditions can improve the robustness of the trained model. To do so, we asked two volunteers to write 26 capital letters, 10 numbers, and 12 gestures in four environments (i.e., High-HL, Low-HL, High-BL, and Low-BL), repeating 30 times per action.



(a) Accuracy. (b) Loss.

Figure 20: Accuracy and loss with different training epochs.

We trained the model with data collected from three environmental conditions and tested with data from the remaining environmental condition. We use 10-fold cross validation to figure out model accuracy. Figure 19 shows the recognition results without our method (w/o US) and with our method (w/ US). Here we show the results of incrementing 10 samples. It delivers that the accuracy of the model trained with data of multiple environmental conditions is about 50%, which is slightly higher than that of the model trained with a single environmental condition (i.e., 40% in Figure 2). This improvement is not enough. Furthermore, this also illustrates the effectiveness and necessity of incremental learning and rejection classification method from another perspective.

6.4. Model Analysis

The Performance of CNN Feature Extraction Module and Different Classifiers. Recall that UltraScr extracts features by training a CNN model and then makes a final decision by an SVM classifier as described in Section 4.3. To illustrate the effectiveness of this CNN-SVM combination model, we compared different classic machine learning classifiers respectively with and without CNN feature extraction module. When building the recognition models without CNN module, we reshape FAP into one-dimensional feature vector used for classification. Figure 18 delivers that the recognition accuracy is about 40% higher with the CNN feature extraction module than without the CNN feature extraction module. This also shows that although the use of the CNN feature extraction module may increase the training time, it can significantly improve the system performance. In addition, by comparing the results of seven classifiers, including Logistics Regression (LR), Support Vector Machine (SVM), k-nearest-neighbor (KNN), Decision Tree (DT), GaussionNB (GNB), Random Forest (RF), and Gradient Boosting (GB), the SVM classifier has higher recognition accuracy than other classifiers, whether with or without CNN feature extraction module.

Impact of Training Epoch. In this experiment, the ratio of the training set to the validation set is 8:2. Figure 20 shows that the accuracy and loss of the training set and validation set are stable when the epoch is 1000. Therefore, we set the epoch as 1000 in UltraScr. Fewer epochs could also improve the model's training speed. The CNN

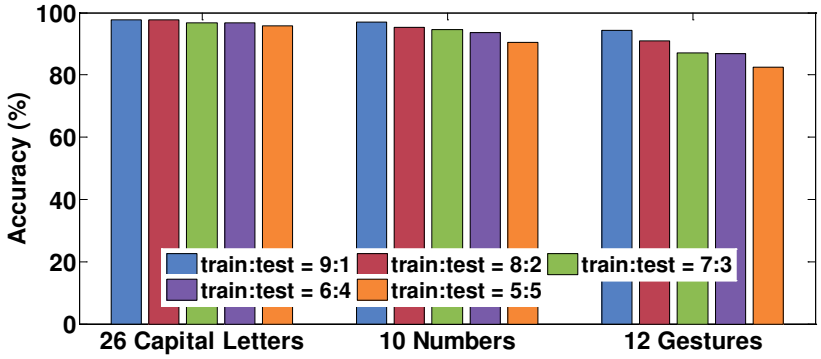


Figure 21: Accuracy of different ratios of training and testing samples.

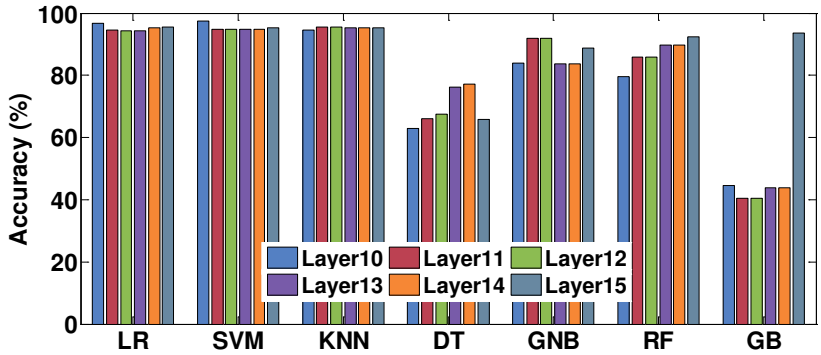
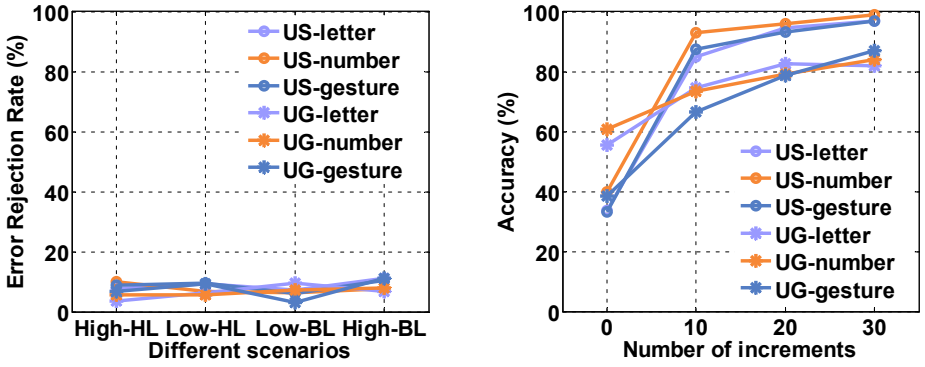


Figure 22: Accuracy when using the signal features extracted by different CNN layers.

model is trained offline, and it only takes 3-4 minutes to complete the training phase on our cloud server. It is easy to notice that the accuracy of the validation set is lower than the recognition accuracy declared by us, mainly because the result here is the result of CNN classification.

Impact of different ratios of training and testing samples. Figure 21 shows the recognition accuracy of 26 capital letters, 10 numbers and 12 gestures under different ratios of training and testing sets. In particular, the total number of training and testing sets for each gesture is 30, and the ratios of training and testing sets are 9:1, 8:2, 7:3, 6:4, and 5:5, respectively. For each ratio setting, we use 10-fold cross validation scheme to calculate the accuracy. Figure 21 shows the results and it delivers that the recognition accuracy decreases as the ratio decrease. This is largely because the larger the ratio, the less the training samples, which results in reducing the robustness of the learned model.

Different Layers. As described in Section 4.3, we conduct feature extraction by training a CNN model. Figure 22 shows the recognition accuracy of extracting different layers of CNN as features when the user writes 26 capital letters on the desktop. The results show that Layer10 has higher recognition accuracy, and the other layers have similar recognition accuracy. The reason behind this is that the higher the number of layers, the more precise the extracted features and the less generalization of the corresponding



(a) Rejection classification method.

(b) Incremental learning.

Figure 23: (a) shows the EER of US and UG when the environment changes; (b) shows the accuracy of US and UG after retraining the model by adding different numbers of samples. Note that US and UG represent using RCM and incremental learning on our approach UltraScr and on UG [21], respectively. Experimental results show that RCM and incremental learning are universal and can be integrated into existing systems with good performance.

model.

Evaluation of New Users. We use 26 capital letters from 13 out of 14 volunteers as the training data set to train the model and the data from another volunteer who do not contribute to the training data set as the test data set. The results show that UltraScr can achieve an accuracy rate of 86% for untrained users. WordRecorder [22] recognizes untrained users at a correct rate of 75%, and it requires users to write on the template in order to ensure that each person writes the same size when collecting data. However, our experiment did not impose any restrictions on users' writing.

6.5. Results of Incremental Learning.

In this experiment, we invite four volunteers, two of them conduct experiments at different humidity levels and the other two conduct experiments at different battery levels. In each experiment, each volunteer writes 26 capital letters, 10 numbers, and 12 gestures on the desktop, repeat 100 times, under two environments, respectively. Note that, to be fair, each operation is repeated 100 times as in UG [21]. Figure 23 shows the results of applying the rejection classification method (RCM) and incremental learning methods to UltraScr (US) and UG [21]. Figure 23a shows the Error Rejection Rate (ERR) of RCM under four kinds of environmental changes. Four kinds of environmental changes include the model trained at 60% Humidity Level (Low-HL) and tested at 90% Humidity Level (High-HL); the model trained at 90% Humidity Level (High-HL) and tested at 60% Humidity Level (Low-HL); the model trained at 90% Battery Level (High-BL) and tested at 10% Battery Level (Low-BL); the model trained at 10% Battery Level (Low-BL) and tested at 90% Battery Level (High-BL). EER refers to the proportion of samples that are falsely rejected among all samples. The lower the EER, the better the RCM performed. As shown in Figure 23a, EER values under four environmental changes are around 10%, whether RCM is used on US or UG. Figure 23b shows that when the environment changes, with the increase in the number of newly added samples,

System	Hardware		Accuracy (%)			Noise Impact	Env. Impact
	Speaker	Mic	Letters	Numbers	Gestures		
WR [22]	0	1	81	-	-	Yes	Yes
Ipanel [24]	0	1	91.2	91.2	91.2	Yes	Yes
UG [21]	1	1/4	-	-	92.75/98.58	No	Yes
UltraScr	1	1	98	95.5	99.62	No	No

Table 3: Comparing our approach against three prior start-of-the arts. We compared the performance of UltraScr with the existing work from four aspects including the number of speakers and microphones, the recognition accuracy, the influence of 60dB noise and the influence of battery & humidity. WR [22] and Ipanel [24] identify letters by recording writing sounds with a microphone, which are easily affected by environmental noise. When the noise exceeds 60dB, the recognition accuracy drops to 50%. UG [21] uses ultrasonic waves to recognize gestures and is not easily affected by environmental noise, but it needs four microphones to achieve 98.58% accuracy. UltraScr can achieve 98% recognition accuracy for 26 capital letters, 95.5% recognition accuracy for 10 numbers, and 99.62% recognition accuracy for 12 gestures using only one speaker and one microphone, and is robust to ambient noise and changes in battery and humidity.

the model's recognition accuracy increases. For US, when the number of newly added samples reaches 10, the recognition accuracy rate is close to 90%. Compared with US, UG recognition accuracy is slightly lower when the same number of new samples is added. The reason behind this is that UG requires a custom device with four microphones, while our experimental device is a commercial mobile phone with two microphones.

6.6. Comparing to Prior Methods.

We compare our system with similar existing ones, including WR [22], Ipanel[24], and UG [21]. Firstly, Table 3 summarizes the performance of different systems. WR and Ipanel recognize handwriting by recording the sound of the hand or pen rubbing against the surface with a microphone, and the recognition accuracy is affected by environmental noise. When the ambient noise reaches 60dB, the recognition accuracy will drop to about 50%. UG proposed a gesture recognition system based on ultrasonic, which can achieve a recognition accuracy rate of 92.75% with one pair of speaker and microphone, but each gesture requires 100 training samples. UltraScr is a context-free gesture recognition system that can recognize letters, numbers, and gestures and is immune to ambient noise because it is based on the characteristics of high-frequency sound waves. In addition, only 30 training samples are required for each gesture. More importantly, UltraScr improves the system's immunity to environmental changes (e.g., battery levels and humidity levels) through RCM-based incremental learning.

Then, we compare UG with the system proposed in this paper (i.e., UltraScr (US)) through an experiment. Note that we do not make an experimental comparison with WR and Ipanel, mainly because the principles behind these systems are different, so there is no comparability. For this experiment, we ask two volunteers to complete 12 gestures (as shown in Table 2) in the air, 100 times each. We use 10-fold cross validation to evaluate UG and US performance in different numbers of training samples. Figure 24 shows that US performs better than UG in different numbers of training samples. In addition, for US, the recognition accuracy of 30 samples for each gesture is basically the same as that of 100 samples for each gesture. However, for UG, at least 80 samples are

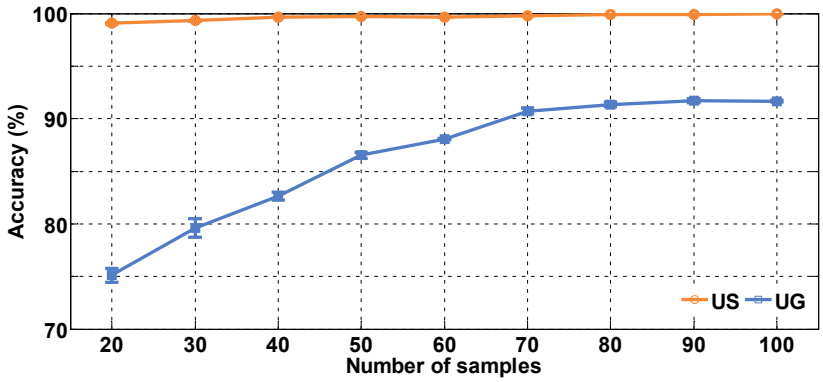


Figure 24: Accuracy of US and UG in recognizing 12 gestures under different number of training samples. 12 gestures are performed in the air. The mean and variance of the 10-fold cross validation are calculated, and the error bar represents the variance.

needed for each gesture to stabilize the recognition accuracy of the model. Furthermore, when the number of training samples is small, the variance of UG is relatively large, indicating that UG needs more samples to increase the robustness of the model. To sum up, the performance of the system proposed in this paper is superior to existing systems in almost all aspects.

7. Discussion and Limitations

In this section, we discuss the limitations of UltraScr.

Need training sets. All methods using machine learning and deep learning require training sets. Compared to other systems, for example, UG [21] requires users to perform each gesture 100 times, our system only requires 30 samples per gesture, which has dramatically reduced the overhead of collecting training sets.

Evaluation of new users. We evaluate a new user using models trained based on other people's training data sets with an accuracy of 86%, which is higher than the current method. In particular, WordRecorder [22] recognizes new users with an accuracy of 75%.

A diverse set of people. UltraScr is currently evaluated only for healthy people. For people with certain diseases, such as Parkinson's, the system's performance does not decrease. The main reason is that illness can slow people down, whereas UltraScr uses signal frequency attenuation to extract gesture characteristics, which is related to multipath but not speed. This is also a significant reason why UltraScr is superior to doppler shift-based methods [45, 46].

8. Related work

Sound-based Letter Recognition: Writing Hacker [47] divides 26 letters into several categories with the same strokes according to different letters and strokes and then infers what each letter is according to the words in the dictionary. The recognition rate can

reach 50%-60%. At the same time, the acceleration sensor is used for correction. However, if there is a continuous burst of noise around, it can not be used properly. When the user speaks continuously, the system can not be used properly. WordRecorder [22] proposes a handwriting recognition system based on sound waves. Users wear a smartwatch on their left hand and write on the desktop with their right hand. The built-in microphone records the voice generated by the friction between the pen tip and paper in the smartwatch. Then a convolutional neural network (CNN) is trained to recognize the capital letters written by users. The average accuracy is 81%. When ambient noise reaches 60 dB, the accuracy rate drops to about 50%. Neither of the two methods can be used when the environment is noisy, primarily when the blasting sound is produced when speaking. Other works [47–50] are all to identify the text written by the user through the recording, which cannot be used in high noise.

Sound-based Gesture Recognition and Location: SoundWave [51], DopLink [31], AirLink [52], Multiwave [46] and AudioGest [45] are sound-based gesture recognition systems using the doppler effect. These systems recognize a small variety of gestures with a broad range of motion. LLAP [20] proposes a method of extracting the received signal phase to calculate the relative distance of the hand movement. Strata [32] estimates the channel of the target object by using the LS channel estimation method and restoring the target's motion estimation on the channel. It also calculates the relative distance change by using phase change. These methods can only estimate the relative distance change by using the LS channel estimation method. Many capital letters cannot be written in one stroke, so this method cannot be used to identify capital letters. AudioGuests [45] uses transmitting signals the same as LLAP to recognize specific gestures by estimating the Doppler shift. With the help of the characteristics of Macbook Air laptop microphone on the left side of the computer rather than in the center of the laptop, the change of the hand's distance to the microphone can be judged by the frequency shift of Doppler shift up and down. If the microphone is in the middle of the computer, it cannot recognize the symmetrical gestures from left to the right and from right to left. C-FMCW [53], ContactlessSleep [54] and FingerIO [55] use absolute position estimation, and can only recognize small movements, such as the heartbeat is moving relative to the rest of the body, but when the body moves, it cannot recognize. Further, all of the location-based systems require at least two pairs of speakers and microphones to restore two-dimensional trajectories.

9. Conclusion

In this paper, we have presented UltraScr, an ultrasound sensing system that recognizes handwritten letters & numbers and common interactive gestures with only one single builtin transceiver pair of the commercial mobile device. UltraScr sends a modulated high-frequency sound wave through the speaker of the mobile device, and the microphone records the echo signal. By analyzing the spectral characteristics of the echo signals, we obtain frequency attenuation profile (FAP) of each operation, then extract the features of each operation through a customized convolutional neural network (CNN) model, and finally identify the operation with a support vector machine (SVM) classifier. Furthermore, we use the incremental learning and rejection classification method (RCM) to reduce the overhead of training data collection and improve the system's robustness

after environment change. We implement UltraScr on mobile phones with the Android system. Since most wearable devices are based on the Android system, UltraScr can be fully used by wearable devices. Our results demonstrate UltraScr can achieve 98% accuracy using only one transceiver pair, across a range of different locations, surface materials, and writing habits.

Acknowledgment

The work was partly supported by the National Natural Science Foundation of China (NSFC) through Grant Agreements No. 61972314, and No. 61872294; in part by the International Cooperation Project of Shaanxi Province (2020KWZ-013, 2019KW-009); and in part by the Shaanxi Province Key R&D Projects (2018SF-369).

References

[1] V. Tran, A. Misra, Q. Roy, K. Choo, Y. Lee, Smartwatch-based early gesture detection & trajectory tracking for interactive gesture-driven applications, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2 (2018) 1–27. doi:10.1145/3191771.

[2] S. Wan, S. Goudos, Faster r-cnn for multi-class fruit detection using a robotic vision system, Computer Networks 168 (2019) 107036. doi:10.1016/j.comnet.2019.107036.

[3] D. Konstantinidis, V. Argyriou, T. Stathaki, G. Nikos, A modular cnn-based building detector for remote sensing images, Computer Networks 168 (2019) 107034. doi:10.1016/j.comnet.2019.107034.

[4] S. Soro, W. Heinzelman, A survey of visual sensor networks, Advances in Multimedia 2009. doi:10.1155/2009/640386.

[5] C. Wang, Y. Wang, Y. Chen, H. Liu, J. Liu, User authentication on mobile devices: Approaches, threats and trends, Computer Networks 170 (2020) 107118. doi:10.1016/j.comnet.2020.107118.

[6] S. Zhang, S. Guo, L. Wang, W. Huang, M. Scott, Knowledge integration networks for action recognition, AAAI 34 (07) (2020) 12862–12869. doi:10.1609/aaai.v34i07.6983.

[7] Q. Wan, Y. Choe, Action recognition and state change prediction in a recipe understanding task using a lightweight neural network model (student abstract), AAAI 34 (10) (2020) 13945–13946. doi:10.1609/aaai.v34i10.7245.

[8] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Computer Networks 54 (15) (2010) 2787 – 2805. doi:https://doi.org/10.1016/j.comnet.2010.05.010.

[9] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, H. Liu, E-eyes: Device-free location-oriented activity identification using fine-grained wifi signatures, in: ACM MobiCom, 2014, pp. 617–628. doi:10.1145/2639108.2639143.

[10] X. Liu, J. Cao, S. Tang, J. Wen, Wi-sleep: Contactless sleep monitoring via wifi signals, Proceedings - Real-Time Systems Symposium 2015 (2015) 346–355. doi:10.1109/RTSS.2014.30.

[11] X. Liu, J. Cao, S. Tang, J. Wen, P. Guo, Contactless respiration monitoring via off-the-shelf wifi devices, IEEE Transactions on Mobile Computing 15 (10) (2016) 2466–2479. doi:10.1109/TMC.2015.2504935.

[12] A. Virmani, M. Shahzad, Position and orientation agnostic gesture recognition using wifi, in: ACM MobiSys, 2017, pp. 252–264. doi:10.1145/3081333.3081340.

[13] J. Zhang, Z. Tang, M. Li, D. Fang, P. Nurmi, Z. Wang, Crosssense: Towards cross-site and large-scale wifi sensing, in: ACM MobiCom, 2018, pp. 305–320. doi:10.1145/3241539.3241570.

[14] Y. Yang, J. Cao, X. Liu, K. Xing, Multi-person sleeping respiration monitoring with COTS wifi devices, in: IEEE MASS, 2018, pp. 37–45. doi:10.1109/MASS.2018.00017.

[15] H. Kong, L. Lu, J. Yu, Y. Chen, L. Kong, M. Li, Fingerpass: Finger gesture-based continuous user authentication for smart homes using commodity wifi, in: ACM MobiHoc, 2019, pp. 201–210. doi:10.1145/3323679.3326518.

[16] C. Jiang, Y. He, X. Zheng, Y. Liu, Orientation-aware rfid tracking with centimeter-level accuracy, in: IEEE/ACM IPSN, 2018, pp. 290–301. doi:10.1109/IPSN.2018.00057.

[17] C. Liu, J. Xiong, L. Cai, L. Feng, X. Chen, D. Fang, Beyond respiration: Contactless sleep sound-activity recognition using rf signals, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 3 (2019) 1–22. doi:10.1145/3351254.

[18] Y. Wang, Y. Zheng, Modeling rfid signal reflection for contact-free activity recognition, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2 (2018) 1–22. doi:10.1145/3287071.

[19] C. Jiang, Y. He, S. Yang, J. Guo, Y. Liu, 3d-omnitrack: 3d tracking with COTS RFID systems, in: ACM/IEEE IPSN, 2019, pp. 25–36. doi:10.1145/3302506.3310386.

[20] W. Wang, A. Liu, K. Sun, Device-free gesture tracking using acoustic signals, in: ACM MobiCom, 2016, pp. 82–94. doi:10.1145/2973750.2973764.

[21] K. Ling, H. Dai, Y. Liu, A. Liu, Ultragesture: Fine-grained gesture sensing and recognition, in: IEEE SECON, 2018, pp. 1–9. doi:10.1109/SAHCN.2018.8397099.

[22] H. Du, P. Li, H. Zhou, W. Gong, G. Luo, P. Yang, Wordrecorder: Accurate acoustic-based handwriting recognition using deep learning, in: IEEE INFOCOM, 2018, pp. 1448–1456. doi:10.1109/INFOCOM.2018.8486285.

[23] O. Saukh, Capturing inhalation efficiency with acoustic sensors in mobile phones, in: the 7th International Workshop, 2018, pp. 19–24. doi:10.1145/3277883.3277889.

[24] M. Chen, P. Yang, J. Xiong, M. Zhang, Y. Lee, C. Xiang, C. Tian, Your table can be an input panel: Acoustic-based device-free interaction recognition, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 3 (2019) 1–21. doi:10.1145/3314390.

[25] L. Lu, J. Yu, Y. Chen, H. Liu, Y. Zhu, L. Kong, M. Li, Lip reading-based user authentication through acoustic sensing on smartphones, IEEE/ACM Transactions on Networking 27 (2019) 447–460. doi:10.1109/TNET.2019.2891733.

[26] X. Xu, J. Yu, Y. Chen, Y. Zhu, L. Kong, M. Li, Breathlistener: Fine-grained breathing monitoring in driving environments utilizing acoustic signals, in: ACM MobiSys, 2019, pp. 54–66. doi:10.1145/3307334.3326074.

[27] M. Zhou, Q. Wang, J. Yang, Q. Li, P. Jiang, Y. Chen, Z. Wang, Stealing your android patterns via acoustic signals, IEEE Transactions on Mobile Computing PP (2019) 1–1. doi:10.1109/TMC.2019.2960778.

[28] M. Zhou, Q. Wang, J. Yang, Q. Li, F. Xiao, Z. Wang, X. Chen, Patternlistener: Cracking android pattern lock using acoustic signals, in: ACM CCS, 2018, pp. 1775–1787. doi:10.1145/3243734.3243777.

[29] A. Valiente, A. Trinidad, J. García-Berrocal, C. Gil, R. ramirez camacho, Extended high-frequency (9-20 khz) audiometry reference thresholds in 645 healthy subjects, International journal of audiology 53 (2014) 531–545. doi:10.3109/14992027.2014.893375.

[30] L. Kinsler, Fundamentals of acoustics, American Journal of Physics 31 (1963) 812–812. doi:10.1119/1.1969118.

[31] M. T. I. Aumi, S. Gupta, M. Goel, E. Larson, S. Patel, Doplink: Using the doppler effect for multi-device interaction, in: ACM UbiComp, 2013, pp. 583–586. doi:10.1145/2493432.2493515.

[32] S. Yun, Y.-C. Chen, H. Zheng, L. Qiu, W. Mao, Strata: Fine-grained acoustic-based device-free tracking, in: ACM MobiSys, 2017, pp. 15–28. doi:10.1145/3081333.3081356.

[33] B. Zhou, J. Lohokare, R. Gao, F. ye, Echoprint: Two-factor authentication using acoustics and vision on smartphones, in: ACM MobiCom, 2018, pp. 321–336. doi:10.1145/3241539.3241575.

[34] A. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, Cnn features off-the-shelf: An astounding baseline for recognition, Vol. 1403, 2014, pp. 512–519. doi:10.1109/CVPRW.2014.131.

[35] L. Liu, P.-l. Yang, W.-w. Sun, J.-w. Ma, Similar handwritten chinese character recognition based on cnn-svm, in: ACM ICGSP, 2017, pp. 16–20. doi:10.1145/3121360.3121376.

[36] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European conference on computer vision, 2014, pp. 818–833. doi:10.1007/978-3-319-10590-1_53.

[37] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (1998) 2278 – 2324. doi:10.1109/5.726791.

[38] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Computer ence.

[39] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Communications of the ACM 60 (6) (2017) 84–90. doi:10.1145/3065386.

[40] J. Wang, K. Zhao, X. Zhang, C. Peng, Ubiquitous keyboard for small mobile devices:harnessing multipath fading for fine-grained keystroke localization, in: ACM MobiSys, 2014, pp. 14–27. doi:10.1145/2594368.2594384.

[41] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Pretten-

hofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, G. Varoquaux, API design for machine learning software: experiences from the scikit-learn project, in: ECML PKDD Workshop: Languages for Data Mining and Machine Learning, 2013, pp. 108–122.

[42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[43] G. Shafer, V. Vovk, A tutorial on conformal prediction, Journal of Machine Learning Research 9 (2007) 371–421. doi:10.1145/1390681.1390693.

[44] Y. Zhao, Z. Qiu, Y. Yang, W. Li, M. Fan, An empirical study of touch-based authentication methods on smartwatches, in: ACM ISWC, 2017, pp. 122–125. doi:10.1145/3123021.3123049.

[45] W. Ruan, Q. Sheng, L. Yang, T. Gu, P. Xu, L. Shangguan, Audiogest: enabling fine-grained hand gesture detection by decoding echo signal, in: ACM UbiComp, 2016, pp. 474–485. doi:10.1145/2971648.2971736.

[46] C. Pittman, P. Wisniewski, C. Brooks, J. J. Laviola, Multiwave: Doppler effect based gesture recognition in multiple dimensions, in: ACM CHI, 2016, pp. 1729–1736. doi:10.1145/2851581.2892286.

[47] T. Yu, H. Jin, K. Nahrstedt, Writinghacker: audio based eavesdropping of handwriting via mobile devices, in: ACM UbiComp, 2016, pp. 463–473. doi:10.1145/2971648.2971681.

[48] G. Luo, M. Chen, P. Li, M. Zhang, P. Yang, Soundwrite ii: Ambient acoustic sensing for noise tolerant device-free gesture recognition, in: IEEE ICPADS, 2017, pp. 121–126. doi:10.1109/ICPADS.2017.00027.

[49] M. Zhang, P. Yang, C. Tian, L. Shi, S. Tang, F. Xiao, Soundwrite: Text input on surfaces through mobile acoustic sensing, in: ACM SmartObjects, 2015, pp. 13–17. doi:10.1145/2797044.2797045.

[50] T. Yu, H. Jin, K. Nahrstedt, Audio based handwriting input for tiny mobile devices, in: IEEE MIPR, 2018, pp. 130–135. doi:10.1109/MIPR.2018.00030.

[51] S. Gupta, D. Morris, S. Patel, D. Tan, Soundwave: using the doppler effect to sense gestures, in: ACM CHI, 2012. doi:10.1145/2207676.2208331.

[52] K.-Y. Chen, D. Ashbrook, M. Goel, S.-H. Lee, S. Patel, Airlink: Sharing files between multiple devices using in-air gestures, in: ACM UbiComp, 2014, pp. 565–569. doi:10.1145/2632048.2632090.

[53] T. Wang, D. Zhang, Y. Zheng, T. Gu, X. Zhou, B. Dorizzi, C-fmcw based contactless respiration detection using acoustic signal, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 1 (2018) 1–20. doi:10.1145/3161188.

[54] R. Nandakumar, S. Gollakota, N. Watson, Contactless sleep apnea detection on smartphones, in: ACM MobiSys, 2015, pp. 45–57. doi:10.1145/2742647.2742674.

[55] R. Nandakumar, V. Iyer, D. Tan, S. Gollakota, Fingerio: Using active sonar for fine-grained finger tracking, in: ACM CHI, 2016, pp. 1515–1525. doi:10.1145/2858036.2858580.