UbiEar: Bringing Location-independent Sound Awareness to the Hard-of-hearing People with Smartphones

SICONG LIU, Xidian University
ZIMU ZHOU, ETH Zurich
JUNZHAO DU, Xidian University
LONGFEI SHANGGUAN, Princeton University
JUN HAN and XIN WANG, Xidian University

Non-speech sound-awareness is important to improve the quality of life for the deaf and hard-of-hearing (DHH) people. DHH people, especially the young, are not always satisfied with their hearing aids. According to the interviews with 60 young hard-of-hearing students, a ubiquitous sound-awareness tool for emergency and social events that works in diverse environments is desired. In this paper, we design UbiEar, a smartphone-based acoustic event sensing and notification system. Core techniques in UbiEar are a light-weight deep convolution neural network to enable location-independent acoustic event recognition on commodity smartphons, and a set of mechanisms for prompt and energy-efficient acoustic sensing. We conducted both controlled experiments and user studies with 86 DHH students and showed that UbiEar can assist the young DHH students in awareness of important acoustic events in their daily life.

CCS Concepts: • **Human-centered computing** → *Ubiquitous and mobile computing systems and tools*;

ACM Reference format:
Sicong Liu, Zimu Zhou, Junzhao Du, Longfei Shangguan, Jun Han, and Xin Wang. 2017. UbiEar: Bringing Location-independent Sound Awareness to the Hard-of-hearing People with Smartphones. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 2, Article 17 (June 2017), 24 pages.
DOI: http://doi.org/10.1145/3090082

1 INTRODUCTION

Sound is an important and subtle way to gain awareness of the surroundings, especially for events out of sight and attention. For example, emergencies such as fire alarms are broadcast to people in homes and public places to notify potential dangers. Household appliances such as microwave ovens beep to indicate that the food is cooked. People knock on doors or ring doorbells to signify their presence to the persons inside.

However, it is non-trivial for the deaf and hard-of-hearing (DHH) people to perceive such important sound awareness. While most DHH people can use visual clues such as facial expressions and sign languages to interpret

Author's address: S. Liu, School of Computer Science and Technology, Xidian University; E-mail: liusc@stu.xidian.edu.cn. Z. Zhou, Computer Engineering and Networks Laboratory, ETH Zurich; E-mail: zzhou@tik.ee.ethz.ch. J. Du, School of Computer Science and Technology, School of Software and Institute of Software Engineering, Xidian University; E-mail: dujz@xidian.edu.cn. L. Shangguan, Department of Computer Science, Princeton University; E-mail: longfeis@cs.princeton.edu. J. Han and X. Wang, School of Software and Institute of Software Engineering, Xidian University; E-mail: {junhan,wangx}@stu.xidian.edu.cn.
Corresponding Author: Junzhao Du.

2474-9567/2017/6-ART17 $15.00
DOI: http://doi.org/10.1145/3090082

speech, they usually rely on bulky assistant infrastructure for *non-speech* sound awareness. For instance, vibration beds are used to communicate tangible alarms during sleep. Wearable displays notify the DHH people by encoding acoustic events into visual signals. Hearing aids and cochlear implants are essential to improve both speech and non-speech sound awareness. Nevertheless, numerous studies [23] [37] report low usage satisfaction with hearing aids due to poor benefit, background noise, fit and comfort problems. A study of adolescents shows that about 38% of cochlear implant users do not wish to be implanted again if their implants failed [45].

In this paper, we aim to design a complementary non-speech sound-awareness tool, especially for young students (primary to high school students). We focus on this group of DHH people because *(i)* students must make daily use of hearing aids at school, thus potentially are more desirable for complementary sound-awareness tools when they feel uncomfortable with hearing aids; and *(ii)* young students tend to be more willing to try new technologies. Our approach is inspired recent ubiquitous computing researches on leveraging smartphones for acoustic sensing and sound awareness [4] [20] [39]. While the basic idea is compelling, a smartphone-based sound-awareness platform for the DHH people needs to fulfil the following requirements:

1. **Location-independent**— The platform should detect and recognize acoustic events accurately in different locations with diverse ambient noises.
2. **Responsive**— The platform should incur low latency and notify the user of the acoustic event promptly.
3. **Energy-efficient**— The platform should consume reasonable energy to enable continuous monitoring on battery-powered smartphones.
4. **User-friendly**— The platform should allow personalized acoustic event subscription and reminder mechanisms and be easy to operate.

Unfortunately, none of the prior works satisfy all these requirements.

In this paper, we propose UbiEar, a mobile platform that provides continuous non-speech sound awareness for the DHH people. UbiEar allows users to subscribe to the acoustic events of interest and choose personalized reminder mechanisms for each acoustic event at a smartphone client. UbiEar contains two components: a cloud server and a local application running on commercial off-the-shelf (COTS) smartphones. A cloud server trains a deep neural network for acoustic event recognition. The smartphone client senses ambient sounds via its microphone, records audio clips that potentially contain an acoustic event of interest, and recognizes the event based on the model downloaded from the cloud server. Finally UbiEar notifies the user promptly via the selected reminder mechanisms.

UbiEar leverages a set of techniques to meet the requirements as a practical sound-awareness platform for the DHH people. To reduce design bias and ensure user-friendly design, we conduct pre-design surveys with 60 hard-of-hearing students to understand the user interaction mechanisms and the acoustic events of interest. For energy efficiency, we design a duty-cycling mechanism to adaptively turn on the microphone. To enable location-independent acoustic event recognition, we adopt a deep convolutional neural network model. The model facilitates to extract less environmental-dependent features for each acoustic event, and is adaptive to locations with diverse ambient noise.

We prototype UbiEar on Android platforms and evaluate its performance through both in-field studies and micro-benchmarks. Experiment results show that UbiEar achieves an average cross-location recognition accuracy of over 90% for 9 acoustic events, which is comparable with the state-of-the-art deep learning model such as CNN [17] and DNN [26], and outperforms representative shallow learning models such as Adaboost [2] and RandomForest [28] by 39% and 18%, respectively. UbiEar incurs an average false positive of 3% and an average missing rate of 4% for acoustic event detection. In addition, UbiEar's neural network model takes ×13 less memory than other deep neural networks for audio sensing [26] [30] and it consumes 16% of the smartphone's battery to run UbiEar continuously for 10 hours.

In the rest of this paper, we review the related work in Section 2, conduct a pre-design survey in Section 3, and detail the system design in Section 4. We next present the system implementation in Section 5, and evaluate the system in Section 6. We finally discuss limitations of this work in Section 7 and conclude in Section 8.

2 RELATED WORKS

Our work is closely related to the following categories of research.

2.1 Assisted Technologies for the DHH People

Hearing aids and cochlear implants are widely adopted tools to enhance sound-awareness for the DHH people. However, users feel uncomfortable and inconvenient to wear hearing aids all the time and everywhere [23] [37] [45]. Other hearing assisted approaches, such as visual [36] [16] and vibration [12] [20] clues are also commonly used for sound awareness. For instance, some researchers propose to assist the DHH people to perceive sound-induced vibration, *e.g.* footsteps, by floor-mounted sensors, or flash lights when the doorbell or telephone rings [12]. Nevertheless, these efforts are restricted to specific sounds and cannot easily be generalized to a wide range of sounds the users might be interested in. UbiEar serves as a complementary approach when hearing aids are inconvenient or uncomfortable to use.

2.2 Hearing Assisted Applications on Mobile Devices

Smartphones and wearable devices also hold promise as sound awareness platforms for the DHH people. Yoo *et al.* [49] design an acoustic recognition algorithm optimized for mechanical sounds to enable wearable sound recognition and notification on smartphones. Ketabdar *et al.* [20] detect changes of ambient acoustic environments to notify users. VisAural [9] is a wearable device that enables sound-localization. Jain *et al.* [16] propose a head-mounted display to visualize and interpret speech. OtoSense [40] is an commercial product that senses acoustic events and translates them into visual and tactile alerts. UbiEar is built upon this thread of efforts, and aims at robust and energy-efficient acoustic event detection and recognition on smartphones.

2.3 Sound Awareness in Unrestricted Environments

Awareness of the acoustic events of interest in unrestricted environments remains a challenge. Some important issues include segmenting successive sounds from an audio stream [14], acoustic event detection in highly noisy environments [43], and multi-source acoustic event separation [48]. Among these issues, how to deal with overlapping acoustic events is particularly important in UbiEar. In [41], the authors propose a supervised sound sound separation method. Researchers have also explored unsupervised sound source separation by using non-negative matrix factorization (NMF) [11] and Independent Component Analysis (ICA) [5]. Mesaros *et al.* [38] propose a generic framework for multi-source acoustic event detection in real life environments. UbiEar adopts techniques from this research area to deal with successive and overlapping acoustic events.

2.4 Machine Learning for Mobile Audio Sensing

Due to the limited computation and storage resources on smartphones, most previous efforts on mobile audio sensing apply shallow learning models such as decision trees [33], linear discriminant classifiers [44] and Gaussian mixture models [34]. However, these models fail to assert their performance in diverse environments [26]. A promising trend is to leverage deep learning models for robust mobile audio sensing. DeepEar [26] enables DNN-based audio recognition on mobile platforms that are equipped with DSPs. To reduce the size of deep learning models to fit in commodity smartphones, DeepX [25] proposes an SVD-based layer compression for the full connection layer. SparseSep [3] designs a sparse coding-based layer compression approach and a convolution kernel separation technique to compress deep neural networks so that they can run on mobile and embedded

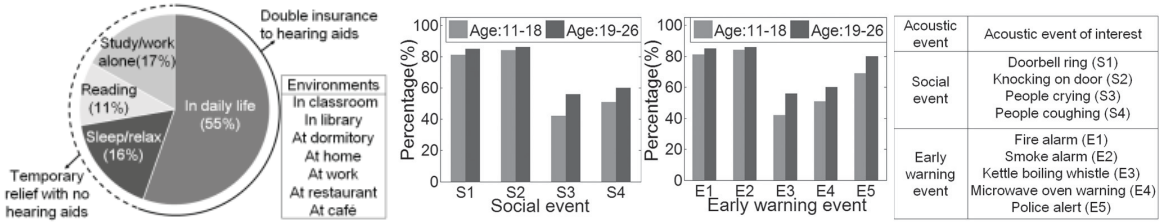
devices. In addition to these compression techniques tailored for mobile audio sensing, other works focus on the compression of more general deep learning models, such as SqeezeNet [13] and deep compression [10]. UbiEar follows this trend of adopting deep learning for mobile audio sensing. It utilizes a light-weight deep learning model to enable robust acoustic event recognition, and only uses the smartphone CPU to operate.

3 DESIGN RESEARCH

To understand the young hard-of-hearing people's need for a smartphone-based sound-awareness tool and their preference of the acoustic events, we design a questionnaire and distribute it to 60 hard-of-hearing students from a school for DHH children in Xi'an, China, with their ages spanning from 10 to 26. We got the permission for conducting the survey from both the school and the interviewees. The participants in our survey are divided into two groups A1 (10 to 18 years old) and A2 (19 to 26 years old). The questionnaire contains both multiple-choice and free-response questions.

3.1 Survey Findings

We briefly summarize the results of our survey as follows.



(a) (b)

Fig. 1. Survey results: (a) Scenarios when alternatives to hearing aids are expected. (b) Popular acoustic events that the hard-of-hearing students are interested in.

3.1.1 Willingness to Use Smartphone-based Sound Awareness Tools. 95% (57/60) students express the needs for a complimentary sound-awareness tool in addition to their hearing aids / cochlear implants that currently in use. 73% (11/25) of group A1 report that the visual clues of ceiling lights installed in the classroom and the dormitory are difficult to remember. 77% (46/60) of the students feel uncomfortable with the vibration beds they use in the dormitory. Overall, 88% (53/60) of students prefer a non-invasive complimentary sound awareness tool that is neither head-mounted nor ear-plugged. Finally 93% (56/60) express willingness to use smartphones for sound-awareness. Figure 1a shows the situations where the students want to use smartphones as a complimentary sound-awareness tool.

3.1.2 Acoustic Events of Interest. While prior works conduct survey on the acoustic events of interest covering a wide range of ages [4] [12], our survey with the young students shows slightly different findings. As shown in Figure 1b, the students are interested in two categories of events, *Social events* and *Early warning events*. Social events are for awareness of the presence or activities of others around, such as doorbell rings (S1), knocking on door (S2), people crying (S3) and people coughing (S4). Early warning events are reminders for the status of appliances such as prompt tone of microwave oven as well as emergencies such as fire and smoke alarms (E1, E2), kettle boiling whistle (E3), microwave oven warning (E4) and police alert (E5).

3.2 Design Goals

To further understand the requirements on the mobile sound-awareness tool, we ask each participant to rate the importance of different performance aspects of a smartphone-based sound-awareness tool and summarize the results as our design goals.

- **Accurate:** 80% (48/60) of the participants report low tolerance for false alarms (no more than twice a day) and 71% (43/60) show no tolerance for miss-detection. 63% (38/60) can tolerate at most three misclassification errors per day.
- **Location-independent:** 70% (42/60) of the participants would like to carry the assistant tool everywhere and it be robust in different environments. Therefore the acoustic event detection and recognition models need to be noise-resilient and adaptive to diverse environments.
- **Responsive:** 70% (42/60) of the participants would tolerate at most six-second delay after the acoustic events of interest occurred. Acoustic event detection and recognition should incur low delay and notify the user promptly.
- **Energy-efficient:** More than 85% (51/60) of the participants express their concerns on the energy consumption of the sound-awareness tool. The acoustic event detection and recognition operation should provide continuous monitoring with low energy consumption.
- **User-friendly:** Only 37% (22/60) of the participants are willing to label and upload audio files to improve the accuracy of sound event recognition. Hence data collection, model training and updating need to be performed automatically with little user intervention.

4 SYSTEM DESIGN

In this section, we start with an overview of UbiEar before elaborating on the design details. Core components of UbiEar include an adaptive duty-cycling sensing mechanism, a sound detection scheme, and a deep learning based sound recognition module. We present the user interface design in Section 5.2.

4.1 Overview

Figure 2 shows the work flow of UbiEar, which consists of five functional modules: adaptive duty-cycling sensing, sound detecting, data pre-processing, sound recognizing, and deep model updating.

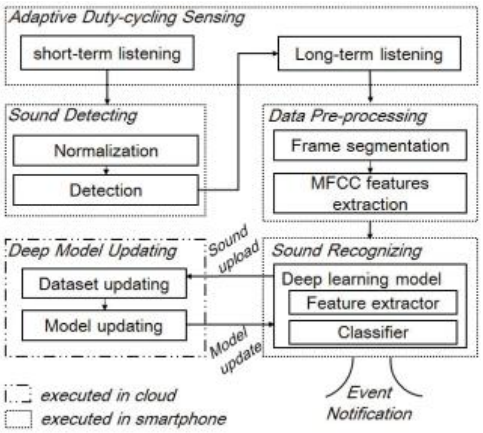


Fig. 2. Work flow of UbiEar.

The **adaptive duty-cycling sensing** module switches between short-term listening and long-term listening. It adopts an adaptive duty-cycling mechanism to balance the energy consumption and the promptness of acoustic event detection. The **sound detecting** module segments the short-duration audio into frames and normalizes the frames. The **data pre-processing** module segments the long-duration audio into frames and extracts the Mel-frequency Cepstral Coefficient (MFCC) features from normalized audio frames. The **sound recognizing** module inputs the MFCC features of long-duration audio frames to a light-weight convolutional neural network, and outputs the inference result. The sound recognizing module is able to learn deep representations for location-independent recognition. The **deep model updating** module collects the audio clips and periodically sends those that have been recognized to the cloud server for retraining and model updating to stay adaptive to environmental changes. We elaborate on each module in sequel.

4.2 Adaptive Duty-cycling Sensing

Since most of the acoustic events of interest occur infrequently and occupy a small portion of duration in daily life, UbiEar adopts an adaptive duty-cycling mechanism to reduce the running time of microphone, and correspondingly, the energy consumption. Specifically, UbiEar switches the microphone among three states: *short-term listening*, *sleeping* and *long-term listening*. In *short-term listening*, the microphone samples audio clips to detect whether there is an acoustic event. If not, it switches to *sleeping* to save energy. Otherwise, it enters *long-term listening* to collect audio clips for acoustic event recognition. To handle successive acoustic events, UbiEar detects the start point, change point and end point of acoustic events from the recorded audio stream (see Section 4.3). The start point and the change point will trigger a thread to input the long-duration ($\leq 2s$) audio clip into the neural network, and the end point will inform the microphone to switch to the sleeping state.

Figure 3 shows an example of the duty-cycling mechanism. After launching UbiEar, the microphone initializes by entering short-term listening. If no acoustic event is detected within the short-term listening state ($[0.0s, 0.2s]$), UbiEar switches into the sleeping state ($[0.2s, 2.0s]$). Since all the events of interest in our evaluation last for at least $2.5s$, we conservatively set the short-term period and the sleeping period to $0.2s$ and $1.8s$ to avoid event missing. If an acoustic event is detected in the short-term listening state ($[2.0s, 2.2s]$), UbiEar switches to long-term listening to capture audio clips. A thread (th2) reads from the audio clips and feeds the audio during $[2.0s, 4.0s]$ into the neural network for feature extraction and event classification (Section 4.4). Thread th1 keeps recording audio clips and detects the end point of this acoustic event (at $2.8s$) and then returns to sleep. Otherwise, if th1 detects a change point, which is likely to be a successive acoustic event (at $6.2s$), then the microphone will keep listening and a new thread th4 is created to read audio clips. Each long-term listening period lasts $2s$ ($[4.2s, 6.2s]$). After collecting audio clips for $2s$, UbiEar extracts MFCC features from the audio clips and feeds them into the neural network to infer a label and the corresponding probability p. If $p < 60\%$, UbiEar sets another long-term listening period, and appends the audio clips to the previous ones for event recognition. UbiEar only returns a label of an acoustic event to the user if $p \geq 60\%$.

4.3 Sound Detecting

The above duty-cycling mechanism relies on a fast and efficient sound detection scheme during short-term listening. Various features from the time and the frequency domains can be used for acoustic event detection [7], and we choose two time-domain features: short time energy (STE) and zero cross rate (ZCR) [32] for their simplicity and effectiveness.

Given an audio clip of $0.1s$, it is divided into frames with an overlap of $1/3$ frame length. Each frame is $0.01s$. Then we calculate the STE and ZCR matrices for each frame and we classify the status of the ith frame $fr(i)$ into

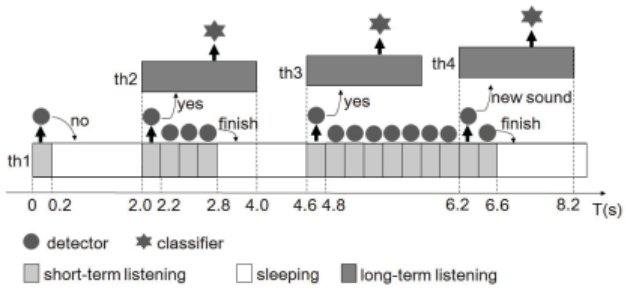


Fig. 3. Illustration of adaptive duty cycling mechanism.

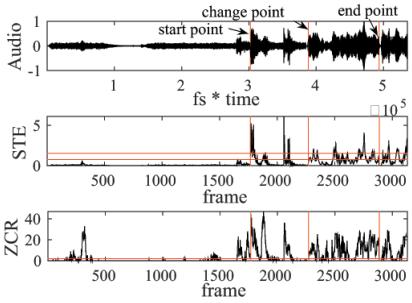


Fig. 4. Sound detection by jointly using ZCR and STE.

silent, *likely-active* and *active* as follows:

$$fr(i) = \begin{cases} \text{silent} & STE(i) < ste_1 \\ \text{likely-active} & ste_1 < STE(i) < ste_2 \text{ or } ZCR(i) > zcr \\ \text{active} & STE(i) \geq ste_1 \end{cases}$$

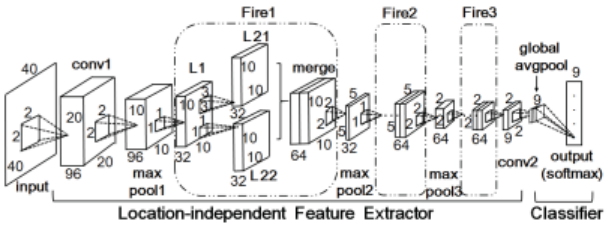
where $STE(i)$ and $ZCR(i)$ are the STE and ZCR of the $i$the frame. $ste_1 = 1/8 * \max\{STE\}$, $ste_2 = 1/4 * \max\{STE\}$, and $zcr = 2$ are three pre-defined thresholds, respectively. The thresholds are dependent on the maximum of STE matrices to be adaptive to different acoustic events. Likely-active frames that are immediately before or after active frames will be regarded as active. Once more than 18 adjacent frames are counted as active, UbiEar decides that certain acoustic event occurs. The parameter of 18 is empirically determined. Among the successive active frames, the location of the first active frame is the start point of the acoustic event, and the last active frame is its the end point. The change point from one acoustic event to another is determined by the Pearson product-moment correlation coefficient [27] of the STE and ZCR matrices between adjacent audio clips.

Figure 4 shows an example of the detection of a knocking-door event (S2), which has a relatively low signal-to-noise ratio among all acoustic events of interest. As is shown, UbiEar accurately detects the start point of this event because the STE is larger than ste_2, the following frames are also detected since the ZCR is more than zcr. A doorbell ring event (S1) occurs after S2, and UbiEar also captures the change point from S2 to S1.

4.4 Sound Recognizing

UbiEar adopts a light-weight convolutional neural network (CNN) for location-independent acoustic event recognition. Most smartphone-based sound awareness systems apply shallow learning algorithms such as decision trees [33] and Gaussian mixture model [34], which fail to extract generic and location-independent features. Conversely, deep learning models such as CNN are fit for non-linear mapping and generalization, making them ideal for location-independent acoustic event recognition. We present in detail the deep learning model used in UbiEar as well as techniques to fit the model into smartphones.

4.4.1 Primer on CNN. A typical CNN consists of an input layer, alternatively multiple convolutional layers and pooling layers, a fully-connected layer and an output layer [17]. The convolutional layers and pooling layers act as a high-quality feature extractor and the fully-connected and output layers resemble a traditional multi-layer perception classifier. The convolutional layers abstract feature maps from input by linear convolutional filters followed by nonlinear activation functions. The number of feature maps is reduced through the pooling layer. The fully-connected layer is used for classification. The feature maps from feature extractor are vectorized by the fully-connected layer followed by a Softmax logistic function.



Layer	operation	output	parameter	Layer	operation	output	parameter
input	—	1*(40*40)	0	Fire1_merge	merge matrix	64*(10*10)	0
conv1	conv filter 96*(2*2)	96*(20*20)	4800 (21.9KB)	...	...	...	...
maxpool1	pool size 2*2	96*(10*10)	0	con2	conv filter 9*(1*1)	9*(2*2)	650 (2.9KB)
Fire1_L1	conv filter 32*(1*1)	32*(10*10)	3104 (14.1KB)	global avgpool	pool size 2*2	9*(1*1)	0
Fire1_L21	conv filter 32*(1*1)	32*(10*10)	1056 (4.8KB)	flatten	flatten array	9	0
Fire1_L22	conv filter 32*(3*3)	32*(10*10)	9248 (42.1KB)	output (softmax)	activation softmax	9	0

Fig. 5. Architecture of the light-weight convolutional neural network used in UbiEar.

4.4.2 Light-weight Neural Network Architecture of UbiEar. Figure 5 illustrates the light-weight CNN model in UbiEar, which also consists of a feature extractor and a classifier. The **location-independent feature extractor** contains two convolutional layers, three Fire modules and three max pooling layers. Each Fire module is comprised of a convolutional layer with $32 \times 1 \times 1$ convolutional filters, and a convolutional layer with a mixture of $32 \times 1 \times 1$ and $32 \times 3 \times 3$ convolution filters. The adoption of 1×1 filters results in fewer parameters while mixing with 3×3 filters contributes to larger activation maps to remain competitive accuracy [13]. Instead of adopting a fully-connected layer for classification, the **classifier** outputs the category confidence of the last Fire module via a global average pooling layer, and then the resulting vector is fed into the output (softmax) layer [29]. Removing the fully-connected layers dramatically reduces the number of parameters and avoid over-fitting [18]. We initialize the number of layers and units by referring to a similar model for audio recognition [26], a 5-layer, 3300-unit DNN model. We then empirically verify the effectiveness of the above parameter settings for audio event recognition. More complex models do not notably improve the recognition accuracy and the current parameter setting fits in the storage and computation capability of the smartphone types used by the our participants.

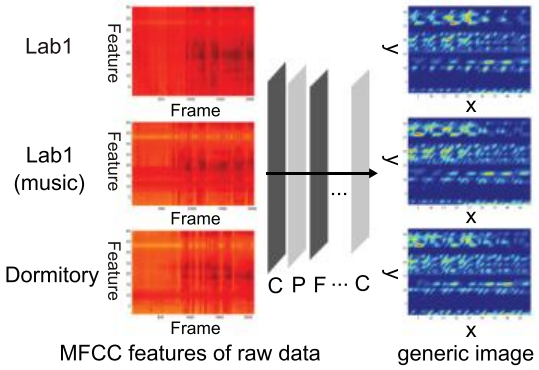


Fig. 6. The generic deep representations of crying sounds are learnt by UbiEar from raw MFCC features in three cases: lab, lab with music, and dormitory. C, P, F represent the convolutional layer, the pooling layer, and the Fire layer, respectively.

Figure 6 visualizes the raw audio clips of crying sounds (Event S3) recorded in three environments (lab, lab with music interference and dormitory) and their deep representations using the above CNN model. As is shown, the raw audio clips differ dramatically while their deep representations are similar and generic. Figure 7 shows the location-independent features which are learnt by the CNN model using the visualization method in [35]. Each color stands for a type of acoustic event and each point is a labeled sample. As shown, the features for different acoustic events are separated in the feature space.

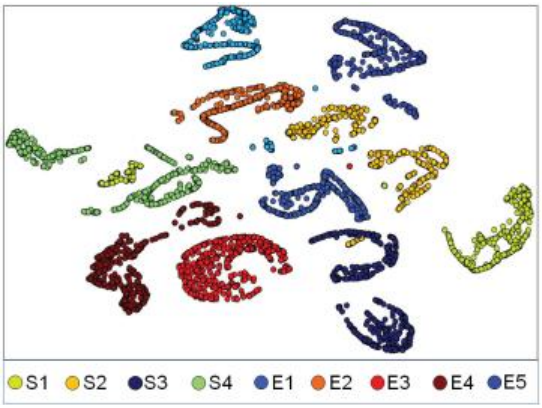


Fig. 7. Visualization of the high-level and location-independent features.

4.4.3 Handling Overlapping Acoustic Events. It is possible that some acoustic events occur almost simultaneously and overlap with each other. In case of overlapped acoustic events, UbiEar outputs either a sequence of detection results where the detection result for each time stamp is the most prominent event, or a multi-strand sequence with multiple simultaneous recognition results, as shown in Figure 8. This can be achieved by adopting unsupervised Non-negative Matrix Factorization (NMF) [11]. The maximum number of simultaneous detection results depends on the number of microphones. After separating overlapped acoustic events, each acoustic event is fed into the acoustic event recognition module for event recognition.

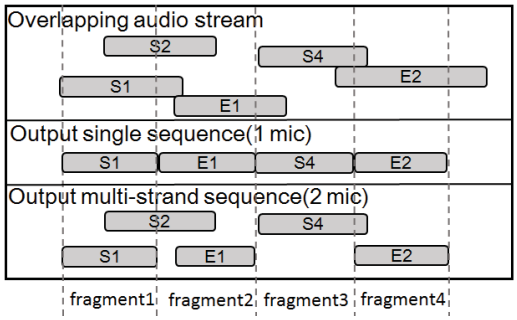
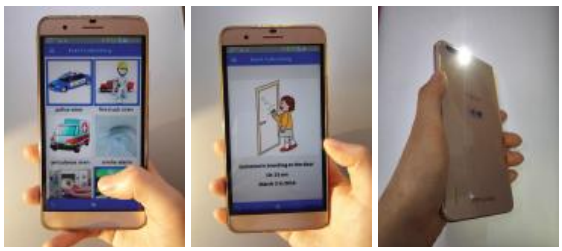


Fig. 8. The output results of overlapping sound from audio stream in different microphones.

4.5 Deep Model Updating

Similar to [26], UbiEar trains the model at the cloud server and downloads the model to smartphones for acoustic event recognition.

To improve the generalization of the deep learning model and adapt to dynamic environments, UbiEar retrains its neural network if newly labeled samples are available. UbiEar updates its neural network in two ways. *(i)* If the label is not in an existing event category, UbiEar conducts dataset augmentation to generate more data samples for this new acoustic event, and retrains all layers of the neural network. *(ii)* If the label belongs to an existing event category, UbiEar evaluates whether the new labeled sample is significantly different from the



(a) (b) (c)

Fig. 9. UbiEar's (a) sound subscribing interface (b) GUI reminder and (c) flash reminder.

existing dataset. Specifically, the new sample is input into both the neural network at the client and a validation model at the server for verification. UbiEar only uses the new labeled sample for retraining if the inference from the models at both the client and the server differ, which indicates that the new labeled sample is different from the training datasets, and the neural network needs updating. In this case, UbiEar retrains the last two layers of the neural network for efficiency. Note that with increasing numbers of newly added audio clips to retrain the model at the server, the model also expands in complexity, *i.e.*, with more layers and parameters to improve accuracy. To ensure that the neural network model can still fit in the storage of smartphones at the client, we further leverage convolution separation and pruning methods such as [30] to generate a compressed neural network model without dramatic precision degradation.

5 IMPLEMENTATION

This section presents the implementation of UbiEar.

5.1 Hardware and Software

The process of model training is conducted in the cloud server, while acoustic event detection and recognition are performed locally on smartphones to avoid intensive wireless data transfer. We prototype the client side of UbiEar on Android 5.0 platform. We implement the signal processing, adaptive duty cycled sensing, event detection, event recognition and event reminder modules in JAVA. UbiEar is also packaged as a JAR to be called by other applications. The sampling rate of the microphone is set to 44.11KHz. Newly recorded audio clip will flush the previous one to retain minimal storage, which is 14KB based on our evaluation. On the server side, we adopt the python library of keras [19] to efficiently train the neural network and use the h5py library [6] to save the trained model. We speed up the model training by performing the computation on NVIDIA Tesla C2050/C2070 GPUs with CUDA 7.5.

5.2 User Interface Design

The user interface of UbiEar consists of event subscribing and event reminder.

Event Subscribing. UbiEar provides a user-friendly acoustic event subscribing interface for hard-of-hearing users. As shown in Figure 9a, each kind of acoustic event is described with a short introduction and a picture. The user is free to choose multiple acoustic events and change them at any time. On detecting any changes in acoustic event subscribing, UbiEar pushes a request of model retraining to the server.

Event Reminder. Visual displaying and vibration are among the top commonly-used and user-friendly sound-awareness display approaches for the hard-of-hearing users [4] [16]. In UbiEar, we offer three options for event reminder (Figure 9), including *(i)* a graphic view with the name of the acoustic event followed by screen flickering,

(ii) phone vibration and *(iii)* flashing of the phone flashlight. Users can select the reminding options for different acoustic events or for different situations (*e.g.* disable flashing if the phone is placed in pocket).

5.3 User Study of User Interface Design

5.3.1 Setups. We conducted a user study in two schools for the deaf and one deaf auditory language rehabilitation center with 86 participants (44 males and 42 females, aging from 10 to 29).

The participants are unaware of the usage of UbiEar beforehand. Participants were instructed to select a native language version of UbiEar on their smartphones, subscribe to their own acoustic events of interest and select the reminder types. Each participant was required to fill in a questionnaire (multiple-choice and free-response questions) regarding the user experience with UbiEar after the studies (see Figure 10).

(a) (b) (c)

Fig. 10. User study of UbiEar and filling questionnaires of the user experience with UbiEar.

5.3.2 Findings on Interface Usage. 87% (75/86) of the participants report that UbiEar is easy to operate. However, some younger participants (10-17 years old) still need guidance from senior students or peer students to instruct them to use UbiEar, as shown in Figure 10. This inspires us to show a usage instruction with a combination of illustrations and words in the system during installation. Among the 86 participants, 84% (73/86) subscribed all the nine acoustic events, which is highly aligned with our pre-design survey. Most participants (63%) reported the notification design is sufficient to remind them of the acoustic events of interest. 70% (60/87) turned on all the three reminder mechanisms for the nine events, while 86% (74/86) choose vibration and 79% (68/86) prefer graphic view as reminder. 24% (21/86) suggested the flickering of screen can be annoying and 28% (24/86) felt that flashing of phone is less useful. Above all, the results suggest that personalized notification is necessary.

6 EVALUATION

This section presents the experimental settings and system performance of UbiEar. We refer to the evaluation methodology in DeepEar [26], the state-of-the-art for mobile audio sensing using deep learning, to comprehensively evaluate the performance of our system.

6.1 Dataset

Table 1 summarizes the datasets used in our evaluation.

UbiSound. We implement a data collection App named UbiSound and recruit 12 volunteers (without hearing losses) to record and label real-world audio clips for two months using UbiSound. The audio clips are collected in 63 locations in Xi'an including homes, offices, supermarkets, streets and so on. As some early warning alarms such as E1 (fire alarm), E2 (smoke alarm) and E5 (police alert) rarely occur during the data collection, we also

If a participant has no smartphone, he/she uses a RedMi 3S phone during the studies by default.

download audio clips of these events from the Internet. The dataset consists of $\approx 565\,hours$, 10, 165, 000 audio chips. It includes 1, 161, 422 clips of S1 (doorbell ring); 1, 172, 501 clips of S2 (knocking on the door), 986, 346 clips of S3 (people crying), 1, 026, 452 clips of S4 (people coughing), 1, 226, 085 clips of E1 (fire alarm), 1, 181, 254 clips of E2 (smoke alarm), 932, 351 clips of E3 (kettle boiling whistle), 1, 096, 528 clips of E4 (microwave oven warning) and 1, 382, 061 clips of E5 (police alert). Each audio clip lasts from 1.5s to 3s.

DCASE 2016. The DCASE 2016 dataset [24] is for a competition of acoustic scene recognition, acoustic event detection in synthetic audio and acoustic event detection in real life audio. We use the given training dataset, which contains 18 clips of S1, 20 clips of S2, and 18 clips of S4.

ESC-50. The ESC dataset [42] is a collection of short environmental audio recordings in a unified format (5-second single channel audio clips sampled at 44.1 kHz). All clips are from public field recordings. The ESC-50 dataset consists of 2000 environmental recordings, which contains 40 clips of S2, 40 clips of S3, and 40 clips of S4.

Table 1. Overview of datasets.

Dataset	Sound type	Total duration	Number of audio clips
UbiSound	S1,S2,S3,S4,E1,E2,E3,E4,E5	565 hours	10,165,000
DCASE 2016	S1,S2,S4	3.7 minutes	54
ESC-50	S2,S3,S4	2.8 hours	120

We further leverage data augmentation techniques [21] to generate 60, 000 more audio clips to enhance the UbiSound dataset. Specifically, we randomly make multiple copies of the audio clips in the UbiSound dataset by changing their volumes. We then mix each copy of audio clip with different ambient noises of different volume. The ambient noises are from the Ambient Noise Dataset [22], which contains 168 ambient sounds. In total the enhanced UbiSound dataset consists of 10, 225, 000 labeled audio clips mixed with 168 environment noises.

6.2 Overall Accuracy

6.2.1 Cross-location Inference Accuracy. We first evaluate the cross-location inference accuracy of UbiEar. Here the accuracy refers to the ratio of the number of correct inferred audio clips over the total number of audio clips tested. To show that UbiEar is location-independent, we train and test UbiEar using audio clips collected from different locations. Specifically, the training datasets consist of all of the data from DCASE 2016 (54 clips), ESC-50 (120 clips), and 80.5% of enhanced UbiSound dataset (8, 231, 125 audio clips collected from 52 locations). We use the remaining 19.5% audio clips of the UbiSound dataset collected from the other 11 places for testing. To simulate the daily life environments of the deaf students, we further collect 9 kinds of typical noises from the homes and the schools of the deaf students including air conditioner, traffic, window opening, showering, water flowing, crowd chatting, raining, computer running, and music, and mix them with the testing dataset.

Figure 11 shows the inference accuracy of UbiEar. The labels along X-axis denotes the 9 kinds of noises we further mixed with the testing dataset. As is shown, UbiEar achieves consistently high inference accuracy for the 9 acoustic events mixed with the 9 kinds of noise. Specifically, UbiEar achieves over 98% accuracy under air-conditioner, music, shower and car horn noises in recognizing E1 (fire alarm), E2 (smoke alarm), E3 (kettle boiling whistle), E5 (police alert) and S1 (doorbell ring). The inference accuracy of S1 (doorbell ring) and S2 (knocking on the door) decreases to 83% when they are mixed with the computer noise and crowd chatting.

We further compare the inference accuracy of UbiEar with four baselines.

- CNN. We use a CNN structure similar to [17], which contains one input layer, two convolutional layers, two pooling layers, one fully-connected layer and one output layer.
- DNN. We adopt the same architecture and parameter sizes as DeepEar [26], which includes one input layer, three fully-connected layers and one output layer.

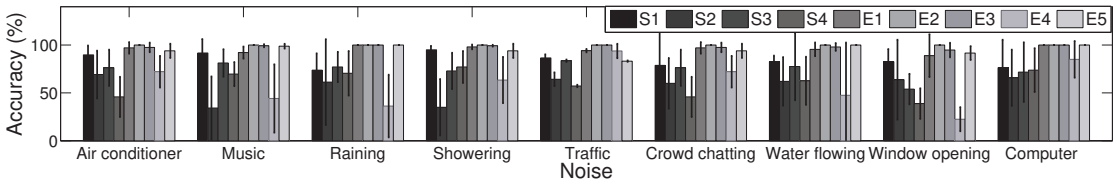


Fig. 11. Cross-location inference accuracy of UbiEar trained by audio clips collected at 52 locations tested on audio clips collected at 11 other locations. The audio clips for testing are further mixed with 9 kinds of noises typical in the daily life of the DHH students. The histogram denotes the mean accuracies, while the error bars are the ranges of mean ± standard deviation.

- Random Forest. We extract 28 MFCC features and then train a random forest classifier that composes of 500 base decision trees as in [28].
- AdaBoost. We extract 28 MFCC features and build a multi-class AdaBoost model as in [2].

Figure 12 shows the inference accuracies (average over the 9 kinds of noises). Compared with the baselines, UbiEar and CNN yield more consistent cross-location inference accuracies over these 9 acoustic events. Specifically, UbiEar significantly outperforms RandomForest in recognizing E1 (fire alarm), E2 (smoke alarm), E5 (police alert). It also achieves higher inference accuracy than DNN, RandomForest and AdaBoost in recognizing S1 (doorbell ring), S2 (knocking on the door), S3 (people crying) and E2 (smoke alarm), E4 (microwave oven warning). The results indicate that UbiEar is more suitable for location-independent acoustic event recognition than the other four baselines.

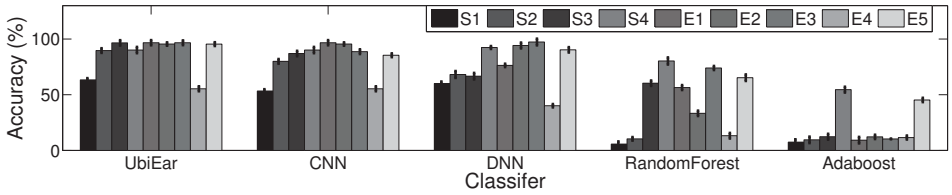


Fig. 12. Comparison of cross-location inference accuracy. The accuracies are averaged over the 9 kinds of noises. The histogram denotes the mean accuracies, while the error bars are the ranges of mean ± standard deviation.

6.2.2 End-to-end Accuracy. In this experiment, we use all the datases in Section 6.1 for training, and recruit 5 volunteers (without hearing losses) to carry UbiEar-installed smartphones to test the end-to-end accuracy of UbiEar (*i.e.* acoustic event detection and recognition) in real-world scenarios. We obtain the ground-truth by video-recording the tests. Each of the 9 acoustic events is tested 100 times in office and home environments. We evaluate the end-to-end accuracy of UbiEar using the following metrics.

- Type 1 error: notifying a non-existent acoustic event, a.k.a. false positive.
- Type 2 error: missing an acoustic event, a.k.a. false negative.
- Type 3 error: misclassifying an acoustic event.

Figure 13 shows the occurrence frequency of type 1, type 2 and type 3 errors. E4 (microwave oven warning) has the most frequent type 1 error (5/100), S3 (people crying) has the most frequent type 2 error (7/100) and type 3 error is most frequently seen (27/100) in E4 (microwave oven warning). In general, type 3 errors occur

more frequently than type 1 and type 2 errors. Type 1 errors are more frequent than type 2 errors in S1 (doorbell ring), S2 (knocking on the door), E1 (fire alarm), E2 (smoke alarm) and E3 (kettle boiling whistle). Due to its short duration, E4 (microwave oven warning) is easily missed and misclassified under loud noises.

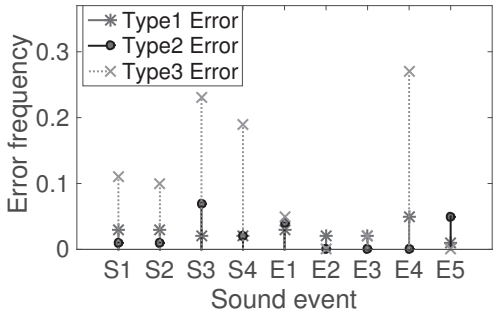


Fig. 13. Occurrence frequency of different types of errors.

To better understand how the acoustic events are misclassified (type 3 errors), we plot the confusion matrix of the 9 acoustic events in Table 2. As is shown, UbiEar achieves more than 90% inference accuracy for S2 (knocking on the door), E1 (fire alarm), E2 (smoke alarm), E3 (kettle boiling whistle) and E5 (police alert). However, S3 (people crying) is easily misclassified as S2 (knocking on door) (5%) and S4 (people coughing) (8%), while E4 (microwave oven warning) is prone to be misclassified as S1 (doorbell ring) (7%), E2 (smoke alarm) (7%) and E3 (kettle boiling whistle) (8%). The cause of low inference accuracy for speech-like events such as S3 (people crying) and S4 (people coughing) are the small amount of training data, while the reason for low accuracy in recognizing E4 (microwave oven warning) is due to low signal-to-noise ratio.

Table 2. Confusion matrix of acoustic event classification (type 3 errors).

	S1	**S2**	**S3**	**S4**	**E1**	**E2**	**E3**	**E4**	**E5**
S1 (doorbell ring)	89	2	4	0	0	0	5	0	0
S2 (knocking on door)	4	90	0	0	0	0	0	6	0
S3 (people crying)	0	5	77	8	0	4	0	0	6
S4 (people coughing)	4	0	5	81	6	4	0	0	0
E1 (fire alarm)	2	0	0	0	95	0	2	0	1
E2 (smoke alarm)	0	0	0	0	0	100	0	0	0
E3 (kettle boiling whistle)	0	0	1	0	0	0	98	1	0
E4 (microwave oven warning)	7	0	0	5	0	7	8	73	0
E5 (police alert)	0	0	0	0	0	0	0	0	100

6.3 System Performance

We then evaluate the system performance including memory usage, energy consumption and system delay, which are crucial for the usability of UbiEar.

Table 3. Comparison of memory usage between UbiEar and recent deep models for mobile devices.

System	Model	Compression Method	Layer Details	Parameters
DeepEar [26]	DNN	None	5 layers: input, fully-conn*3, output	2,700,299 (10.8M)
Deep Compression [10]	DNN	pruning	5layers:input,fully-conn*3,output	300,034 (1.2M)
DeepX [25]	DNN	SVD	5layers:input,fully-conn*3,output	1,955,391(7.8M)
SparseSep-DNN [3]	DNN	sparse coding	5layers:input,fully-conn*3,output	1,033,680 (4.1M)
MobiEar [31]	CNN	N/A	7 layers:input, conv*2, maxpool*2,fully-con, output	430,955 (1.6M)
SparseSep-CNN [3]	CNN	sparse coding and convolution separation	7 layers:input, conv*2, maxpool*2, fully-con, output	264,972 (0.98M)
SCNN [30]	CNN	convolution separation	10layers:input,conv*5,fully-conn*2,dropout,output	2,543,097 (10.2M)
DeepPed [47]	CNN	convolution pruning	10layers:input,conv*5,fully-conn*2,dropout,output	635,775 (2.6M)
UbiEar	CNN	no fully-conn layer, using 1*1 convolutional filters	14 layers: input, conv*2, Fire*3, maxpool*3, global avgpool, flatten, output	43,691 (199KB)

6.3.1 Memory Usage. Table 3 compares the memory usage of the neural network in UbiEar with other models that are intended for mobile devices. As is shown, DNN based models [3] (DNN version) [10] [26] [25] usually contain notably more parameters because of the fully-connected layers. In contrast, CNN based models [3] (CNN version) [30] [31] [47] can achieve comparable accuracy with fewer parameters and thus less memory usage. In addition, our CNN-based UbiEar takes only 199KB of memory.

6.3.2 System Delay. Figure 14 plots the delays of event detection and recognition (extraction features and recognition). In this evaluation, the model parameters are loaded in the smartphone memory, and the neural network is run using smartphone CPU. For each event, we run UbiEar on 50 randomly picked audio clips and collect the processing delays of sound detection and recognition modules from the log files of Android Studio. The delays in Figure 14 are averaged over 50 runs. The major delay comes from time-domain feature extraction in event detection and event recognition (including MFCC extraction and running neural network models). As shown in Figure 14, all acoustic events can be recognized within 40ms. Specifically, UbiEar can determine whether an acoustic event of interest occurs within 20ms. On detecting an acoustic event, UbiEar samples 2s of audio clip as input for sound recognition (MFCC feature extraction, deep representation extraction and sound classification by the neural network) and the recognition time is within 20ms.

To further reduce the delay of UbiEar, we implement a mechanism to load the neural network from the smartphone cache. As discussed above, the neural network in UbiEar occupies only 199KB, which can be fit into the memory cache of commodity smartphones. We adopt LruCache [8] to implement the cache access and management mechanism, which keeps recently referenced objects in a strong reference and evicts the least recently used object before the cache overflows. Loading the neural network from the cache is $\times$1.8 faster than invoking the neural network from memory, and $\times$2.5 faster than from the SD card.

6.3.3 Energy Consumption. We run UbiEar on an RedMi 3S smartphone to measure the power consumption. The battery capacity of the RedMi 3S smartphone is 4000 mAh. In UbiEar, energy is mainly consumed by operations including microphone sampling, acoustic event detection, event recognition and event notification. During the measurement, only UbiEar and the system Apps of Android OS are run on the smartphone. We manually generate 9 kinds of acoustic events at random during 10 hours. In total there are 100 (For S1-S4, each event is generated 10 times and for E1-E5, each event is generated 12 times) acoustic events. As an illustration, Figure 15 shows the energy consumption of UbiEar during the first hour. Each bar represents the energy consumed by UbiEar at the granularity of 10 minutes within the hour. There are 3, 1, 1, 1, 2 and 1 acoustic events during each of the 10-minute duration, respectively. The energy consumption is calculated using the testing logs (the battery historian charts) from the adb tool of Android Studio [46]. The log records the energy consumption from the microphone sampling, CPU computation, the screen, the vibrator and the camera. Due to the adaptive duty cycling mechanism, during the 10-hour measurement, the microphone only samples audio clips for around 1.35 hours in total. According to the running log, UbiEar executes event detection modules 17695 times, loads deep

Table 4. Comparison of UbiEar with other commercial sound awareness projects.

Project	Sound Type	Recognizer	Accuracy	Memory	Delay	Energy	Usage	Devices	Client/Server Architecture
OtoSense [40]	machine vibrations, alarms, gunshots, vehicles, steps	deep learning	95%	60-140M	5s	≈ 9.87mAh per hour	normal	smartphone/ others	inference at client (edge devices)
Audio Analytic [1]	people crying, alarms, window break, aggressive voice	machine learning	94%	compact	––	low	smart home, baby monitor, elderly care	software sensors	inference at server
Leeo Smart Alert [15]	smoke and CO alarms	––	––	––	––	plugs into wall outlet	home protect	special devices, Wi-Fi	inference at server
UbiEar	social events& early warning events	deep learning	95%	60-100M	< 50ms	< 1.3mAh per recognition	DHH people	smartphones	inference at client (phones)

learning models for event recognition 126 times, and triggers user notification 103 times. Therefore, the energy consumption of UbiEar during this 10-hour measurement is 16% of the battery capacity. In comparison, when the microphone samples audio continuously for ten hours with the screen turned off, UbiEar costs about 13% of the battery capacity. Given a fully charged RedMi 3S smartphone, we continuously run UbiEar with acoustic events occurring at an average frequency of 10 time per hour. We run the test for three times during 12 days, and the smartphone can run for 48 to 51 hours before the battery drains.

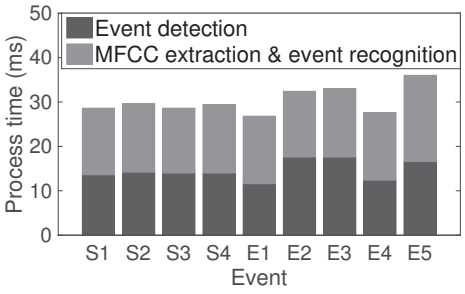


Fig. 14. System delay.

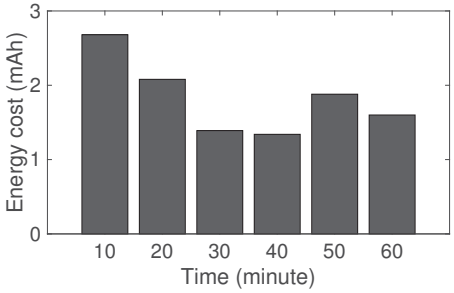


Fig. 15. Energy consumption.

6.3.4 Comparison with Commercial Systems. Table 4 summarizes the main characteristics of UbiEar compared with three other commercial sound-awareness projects. Otosense [40] is a smartphone based sound awareness project. Audio Analytic [1] and Leeo Smart Alert [15] require special devices such as software sensors and nightlight, and are used for smart home applications. In contrast, UbiEar requires no special devices, and conducts acoustic event recognition on the client side without requiring Wi-Fi/cellular networks to connect with the cloud while achieving comparable inference accuracy.

6.4 Micro-benchmarks

This thread of experiments evaluates the impact of various parameters on the accuracy of UbiEar. Similar to the evaluations in Section 6.2.1, we use roughly 80% audio clips of the enhanced UbiSound dataset for training, and the remaining 20% for testing.

6.4.1 Short-term Sensing Duration. During short-term sensing, the microphone collects audio to detect whether there is an acoustic event of interest. It is observed that the detection accuracy changes with the short-term sensing duration (see Figure 16a). The accuracy is above 85% if the short-term sensing duration is longer than $0.2s$. The highest accuracy is seen with a short-term sensing duration of $0.225s$ and $0.25s$. Since the short-term

sensing duration of 0.2*s* results in the highest median accuracy, we set the default short-term sensing duration to 0.2*s*.

6.4.2 Threshold of ZCR. As discussed in Section 4.3, we use two time-domain features, ZCR and STE, for acoustic event detection. The threshold of STE is adaptive to the maximum of the recorded audio clip. The threshold of ZCR needs to be optimized empirically. Figure 16b shows that the detection accuracy using various thresholds of ZCR. We set the threshold of ZCR to 3, which achieves the highest median detection accuracy over all the 9 acoustic events.

6.4.3 Long-term Sensing Duration. The microphone enters long-term sensing status to sample audio for the neural network model. Figure 16c shows the impact of long-term sensing duration on recognition accuracy. UbiEar has the similar recognition accuracy over 9 kinds of acoustic events when using a long-term sensing duration of 1.5*s*, 1.75*s*, 2*s* and 2.25*s*. However, 2*s* leads to the highest median accuracy of 88%. Therefore we set the long-term sensing duration as 2*s*.

6.4.4 Number of Filters in MFCC. The number of filters in extracting MFCCs affects the energy consumption in audio event detection. We evaluate the impact of the number of filters used to extract MFCCs on the recognition accuracy in Figure 16d. As is shown, UbiEar achieves accuracies of above 70% for most kinds of acoustic events using 40 and 50 filters. To save energy, we configure the number of filters to be 40.

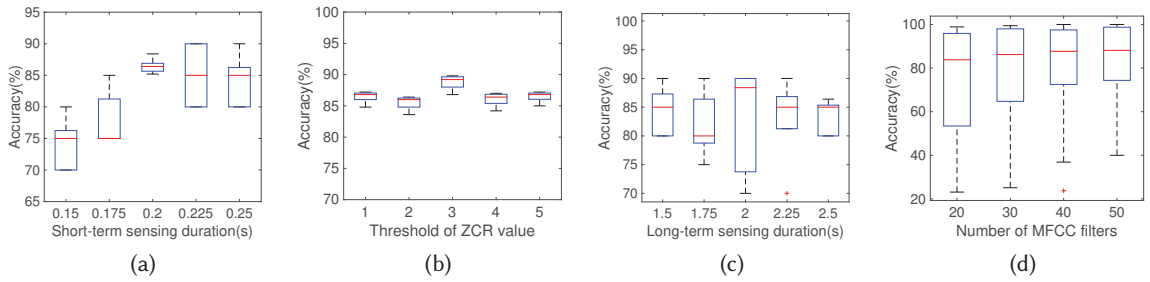


Fig. 16. Impact of (a) short-term sensing duration and (b) ZCR threshold on acoustic event detection accuracy; Impact of (c) long-term sensing duration and (d) number of MFCC filter on acoustic event recognition accuracy.

6.4.5 Number of Training Samples. To evaluate how the size of the training dataset affects the cross-location inference accuracy, we randomly choose 10% to 100% of the training dataset for training. Figure 17 shows the inference accuracies for each acoustic event using different amount of training samples. We find that the accuracies for most early warning events such as E1 (fire alarm), E2 (smoke alarm), E3 (kettle boiling whistle) can reach 90% using only 40% of the training data. However, some social events such as S3 (people crying) and S4 (people coughing) require 100% of the training data to achieve high accuracy. This is because the sounds of early warning events *e.g.* E1 (fire alarm) usually have fixed patterns (*e.g.* the same frequency), while the sounds of social events *e.g.* S2 (knocking on the door) can vary dramatically on different doors and by different people. Thus social events often need a more diverse training dataset to achieve satisfactory accuracy.

6.4.6 Number of Iterations in the CNN Model. Figure 18 shows the accuracies of UbiEar using different numbers of iterations by conducting 10-fold cross-validation. We find that 8000 iterations achieve high and balanced accuracies for all the 9 acoustic events, which we used during our performance evaluation.

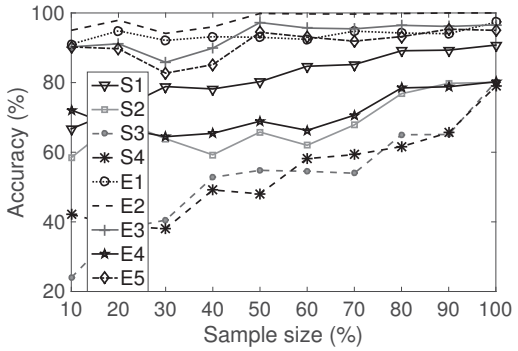


Fig. 17. Impact of size of training dataset.

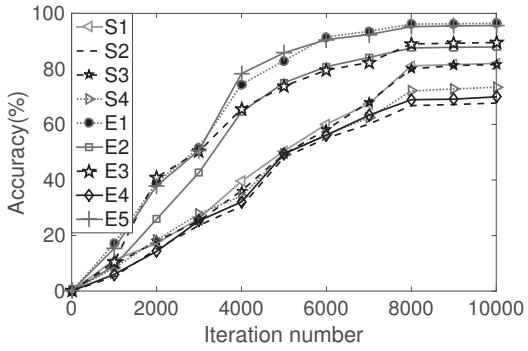


Fig. 18. Impact of iteration numbers.

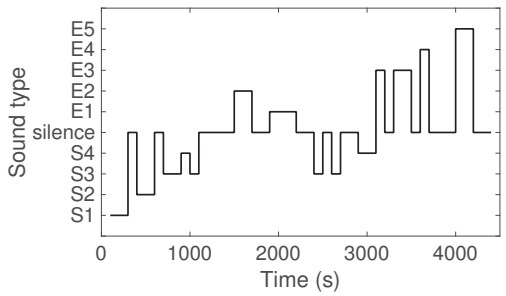


Fig. 19. Sound detection sequence.

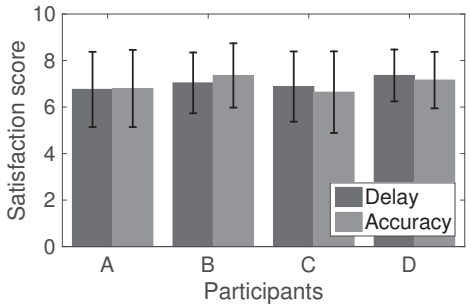


Fig. 20. Satisfaction scores of accuracy and delay.

6.5 Case Studies

Similar to our study on user interface design, we deploy UbiEar to the 86 participants from two schools for the deaf and one deaf auditory language rehabilitation center, and conduct a 2-day experiment during the school time (09:00 to 17:00). We are permitted to video-record the reactions of the participants to collect ground truths of acoustic events. We do not artificially generate audio clips for social and some of the early warning events. However, since emergency events, *e.g.* fire alarms, rarely occur in daily life, we simulate these events by playing recorded alarms. After the 2-day experiment, all the 86 participants fill in a questionnaire regarding the satisfaction of response delay and system accuracy. In the questionnaire, we distinguish 4 cases of the participants: Case A: hard-of-hearing with hearing aids (34/86), Case B: hard-of-hearing without hearing aids (16/86); Case C: deaf with hearing aids (27/86), and Case D: deaf without hearing aids (9/86).

Figure 19 shows an example trace segment of the acoustic events detected during the study with 14 acoustic events. In terms of type 1 errors, the non-existent doorbell ring alarm (S1), knocking door (S2) and people crying (S3) were notified once, twice, and once, respectively. As for type 2 errors, three acoustic events, S3 (people crying), S4 (people coughing) and E3 (kettle boiling whistle), are missed once, once and three times, respectively because of the low volume of the acoustic events. For type 3 errors, E4 (microwave oven warning), S3 (people crying), and S4 (people coughing) are easily misclassified. There are totally eight times of misclassification errors. The lowest recognition accuracy is seen for S4 (people coughing), S3 (people crying) and E4 (microwave oven warning), because they are often in low volume and easily overwhelmed in highly noisy environments.

Figure 20 summarizes the user satisfaction score (0-10) on the response delay and recognition accuracy. Score 0 is the most unsatisfied while store 10 is the most satisfied. 82.5% (71/86) of the participants report that UbiEar is responsive in reminding all the acoustic events of their interests. 75.6% (65/86) of participants report a higher tolerance for extra notification (type 1 errors) than for missing events (type 2 errors) and misclassification (type 3 errors). For participants in Case A and Case C, who wear hearing aids, 70.6% (24/34) of the participants in Case A and 58.3% (16/27) of those in Case C report the need for more accurate notification (acoustic event detection). For participants in Case B and Case D, who do not wear hearing aids during the study, 31.0% (5/16) of of the participants in Case B and 33.3% (3/9) of those in Case D report the need for more accurate notification (acoustic event detection). It is observed that the students with a higher level of hearing loss and those without wearing a hearing aid demand more prompt and accurate notification because they tend to rely more on the sound-awareness tool. 58.8% (20/34) of the participants in Case A, 51.8% (14/27) in case B, 56.3% (9/16) in Case C and 33.3% (3/9) in Case D state their medium (4-6) satisfaction to more than three numbers of the extra notifications (type 1 errors). 73.5% (25/34) of the participants in Case A, 66.7% (18/27) in case B, 62.5% (10/16) in Case C, and 55.6% (5/9) in Case D report their high satisfaction (7-10 scores) to the missing rate of event detection (type 2 errors). 88.2% (30/34) of the participants in Case A, 85.1% (23/27) in case B, 87.5% (13/16) in Case C, and 77.8% (7/9) in Case D express their high satisfaction (7-10 scores) to the low event classification errors (type 3 errors). 39.5% (34/86) of the participants remark that short-lasting events such as S4 (people coughing) and D3 (people crying) need more accurate notification.

7 DISCUSSION

There are several limitations of our work that may need future exploration.

7.1 Variations in Acoustic Events

Although UbiEar adopts a deep learning model, which is trained on rich datasets to accurately distinguish 9 kinds of acoustic events, we find that some participants in the user studies still desire a more accurate sound-awareness tool. Variations in the acoustic events still remain a challenge for general audio sensing. For instance, one participant reports low inference accuracy in S2 (knocking on the door) because one of his friends simply knocks on the door softly for just once. This variation of S2 only takes up a limited portion of the training dataset, which leads to erroneous inference results. To deal with the new variations of acoustic events, users can upload the audio clips of the new variations to the server for model updating (see Section 4.5). The challenge is to design a way for the DHH users to label the new variations. One possibility is to upload the unlabeled audio clips and crowdsource the labeling tasks to users without hearing losses, which we leave for future work.

7.2 Acoustic Events in Unconstrained Environments

UbiEar applies a preliminary mechanism to deal with successive and overlapped acoustic events when detecting acoustic events. However, as discussed in Section 2, the robustness and capability of the mechanism is dependent on the number of microphones available and the signal-to-noise ratio. In addition, UbiEar only classifies the audio clip into one of the nine categories of events. In unconstrained environments, there can be other acoustic events that do not belong to any of these events. During our user study, several young students deliberately knock on the desk to mimic knocking on the door, which causes interference to UbiEar and decreases the accuracy of S2 (knocking on the door). While it is impossible to account for all these interferences in unconstrained environments, it is beneficial to investigate the potential environments where the sound-awareness tools might be used and identify the interferences beforehand. An alternative we plan to explore is to integrate sound localization as a filter to sieve unexpected acoustic events. For instance, the knocking sound from a desk and the door may

travel different distances and thus can be distinguished. Sound localization might also be helpful to filter out sounds from radios and TVs.

7.3 Evaluations on Model Updating

To improve the robustness of UbiEar in terms of variations of acoustic events and unconstrained environments, one solution is to update the model when necessary. We have proposed a model updating scheme in Section 4.5. However, due to the limited numbers of new audio clips that are significantly different from our training dataset, we did not evaluate the model updating mechanism in this work. We anticipate the use of large-scale datasets for model training, testing and updating.

7.4 Extending to Other User Groups

We motivate our work by conducting surveys with DHH students. As we have already noticed, the user needs of this user group is slightly different from the findings based on other user groups [4] [12]. It can be non-trivial to extend our smartphone-based approach to other user groups. For instance, the elderly might not use smartphones frequently, and UbiEar needs to be ported to other mobile devices, which may have limited storage, battery and computation power. The user interface design may also be tuned for other user groups.

8 CONCLUSION

In this paper, we propose UbiEar, a smartphone-based sound-awareness system for the young hard-of-hearing people. It leverages a deep learning architecture for location-independent acoustic event recognition, a core need for young hard-of-hearing people who cannot afford or are unsatisfied with their hearing aids. We also design a set of mechanisms to enable deep learning based acoustic sensing on computation and energy constrained smartphones. The design of UbiEar is guided by pre-design surveys with 60 hard-of-hearing participants, and evaluated by both controlled experiments and case studies with 86 hard-of-hearing participants. Experimental results show that UbiEar outperforms conventional shallow learning based acoustic sensing systems in accuracy, while retaining satisfactory system delay and energy efficiency.

ACKNOWLEDGMENTS

This work is partially supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61502374, 61472312, and 61272456; the Fundamental Research Funds for the Central Universities under project No. BDY041409 and JB151002 (Xidian University); and the CETC shining Star Innovation.

REFERENCES

[1] Audio Analytic. 2017. https://www.audioanalytic.com/. (2017).
[2] James Bergstra, Norman Casagrande, Dumitru Erhan, Douglas Eck, and Balázs Kégl. 2006. Aggregate features and AdaBoost for music classification. *Machine Learning* 65, 2-3 (2006), 473–484.
[3] Sourav Bhattacharya and Nicholas D Lane. 2016. Sparsification and Separation of Deep Learning Layers for Constrained Resource Inference on Wearables. In *Proc. SenSys*. ACM, 176–189.
[4] Danielle Bragg, Nicholas Huynh, and Richard E Ladner. 2016. A Personalizable Mobile Sound Detector App Design for Deaf and Hard-of-Hearing Users. In *Proc. ASSETS*. ACM, 3–13.
[5] J-F Cardoso. 1998. Multidimensional independent component analysis. In *Proc. ICASSP*, Vol. 4. IEEE, 1941–1944.
[6] Andrew Collette. 2015. HDF5 for Python. http://www.h5py.org/. (2015).
[7] Simon Dixon. 2006. Onset detection revisited. In *Proc. DAFx*.
[8] Google. 2016. android.util.LruCache. https://developer.android.com/reference/android/util/LruCache.html. (2016).
[9] Benjamin M Gorman. 2014. VisAural:: a wearable sound-localisation device for people with impaired hearing. In *Proc. ASSETS*. ACM, 337–338.

[10] Song Han, Huizi Mao, and William J Dally. 2016. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *Proc. ICLR*.
[11] Toni Heittola, Annamaria Mesaros, Tuomas Virtanen, and Antti Eronen. 2011. Sound event detection in multisource environments using source separation. In *Proc. CHiME*. 36–40.
[12] F Ho-Ching, Jennifer Mankoff, and James A Landay. 2003. Can you see what i hear?: the design and evaluation of a peripheral sound display for the deaf. In *Proc. CHI*. ACM, 161–168.
[13] Forrest N Iandola, Matthew W Moskewicz, Khalid Ashraf, Song Han, William J Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 1MB model size. *arXiv preprint arXiv:1602.07360* (2016).
[14] Keisuke Imoto and Nobutaka Ono. 2015. Acoustic scene analysis from acoustic event sequence with intermittent missing event. In *Proc. ICASSP*. IEEE, 156–160.
[15] Leeo Inc. 2017. Meet the Leeo Smart Alert Nightlight. https://www.leeo.com/meet-the-leeo-smart-alert-nightlight/. (2017).
[16] Dhruv Jain, Leah Findlater, Jamie Gilkeson, Benjamin Holland, Ramani Duraiswami, Dmitry Zotkin, Christian Vogler, and Jon E Froehlich. 2015. Head-Mounted Display Visualizations to Support Sound Awareness for the Deaf and Hard of Hearing. In *Proc. CHI*. ACM, 241–250.
[17] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *Proc. MM*. ACM, 675–678.
[18] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-scale video classification with convolutional neural networks. In *Proc. CVPR*. IEEE, 1725–1732.
[19] Keras. 2016. Keras: Deep Learning library for Theano and TensorFlow. https://keras.io/. (2016).
[20] Hamed Ketabdar and Tim Polzehl. 2009. Tactile and Visual Alerts for Deaf People by Mobile Phones. In *Proc. ASSETS*. ACM, 253–254.
[21] Adel Khalil, James Sun, Yu Zhang, and Gordon Poole. 2014. RTM noise attenuation and image enhancement using time-shift gathers. In *Proc. EAGE Conference and Exhibition*.
[22] George Kimura. 2017. Ambient Noise Database. http://www.ntt-at.com/product/noise-DB/. (2017).
[23] Sergei Kochkin. 2000. MarkeTrak V: Why my hearing aids are in the drawer: The consumers' perspective. *The Hearing Journal* 53, 2 (2000), 34–36.
[24] Gregoire Lafay. 2017. IEEE DCASE 2016 Challenge-Task 2-Train/Development Datasets. https://archive.org/details/dcase2016_task2_train_dev. (2017). Published February 10, 2016.
[25] Nicholas D Lane, Sourav Bhattacharya, Petko Georgiev, Claudio Forlivesi, Lei Jiao, Lorena Qendro, and Fahim Kawsar. 2016. DeepX: A software accelerator for low-power deep learning inference on mobile devices. In *Proc. IPSN*. ACM, 1–12.
[26] Nicholas D Lane, Petko Georgiev, and Lorena Qendro. 2015. DeepEar: robust smartphone audio sensing in unconstrained acoustic environments using deep learning. In *Proc. UbiComp*. ACM, 283–294.
[27] Joseph Lee Rodgers and W Alan Nicewander. 1988. Thirteen ways to look at the correlation coefficient. *The American Statistician* (1988), 59–66.
[28] Andy Liaw and Matthew Wiener. 2002. Classification and regression by randomForest. *R news* 2, 3 (2002), 18–22.
[29] Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. *arXiv preprint arXiv:1312.4400* (2013).
[30] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Pensky. 2015. Sparse convolutional neural networks. In *Proc. CVPR*. IEEE, 806–814.
[31] Sicong Liu and Junzhao Du. 2016. Poster: MobiEar-Building an Environment-independent Acoustic Sensing Platform for the Deaf using Deep Learning. In *Proc. MobiSys*. ACM, 50–50.
[32] Nitin N Lokhande, Navnath S Nehe, and Pratap S Vikhe. 2012. Voice activity detection algorithm for speech recognition applications. In *Proc. ICCIA*. IJCA.
[33] Hong Lu, Wei Pan, Nicholas D Lane, Tanzeem Choudhury, and Andrew T Campbell. 2009. SoundSense: scalable sound sensing for people-centric applications on mobile phones. In *Proc. MobiSys*. ACM, 165–178.
[34] Hong Lu, Jun Yang, Zhigang Liu, Nicholas D Lane, Tanzeem Choudhury, and Andrew T Campbell. 2010. The Jigsaw continuous sensing engine for mobile phone applications. In *Proc. SenSys*. ACM, 71–84.
[35] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* Nov (2008), 2579–2605.
[36] Tara Matthews, Janette Fong, and Jennifer Mankoff. 2005. Visualizing non-speech sounds for the deaf. In *Proc. ASSETS*. ACM, 52–59.
[37] Abby McCormack and Heather Fortnum. 2013. Why do people fitted with hearing aids not wear them? *International Journal of Audiology* 52, 5 (2013), 360–368.
[38] Annamaria Mesaros, Toni Heittola, Antti Eronen, and Tuomas Virtanen. 2010. Acoustic event detection in real life recordings. In *Proc. EUSIPCO*. IEEE, 1267–1271.
[39] Matthias Mielke and Rainer Brueck. 2015. Design and evaluation of a smartphone application for non-speech sound awareness for people with hearing loss. In *Proc. EMBC*. IEEE, 5008–5011.
[40] OtoSense. 2017. https://www.otosense.com/. (2017).

[41] Jouni Paulus and Tuomas Virtanen. 2005. Drum transcription with non-negative spectrogram factorisation. In *Proc. EUSIPCO*. IEEE, 1–4.
[42] Karol J Piczak. 2015. Environmental sound classification with convolutional neural networks. In *Proc. MLSP*. IEEE, 1–6.
[43] Ilyas Potamitis, Stavros Ntalampiras, Olaf Jahn, and Klaus Riede. 2014. Automatic bird sound detection in long real-field recordings: Applications and tools. *Applied Acoustics* 80 (2014), 1–9.
[44] Tauhidur Rahman, Alexander Travis Adams, Mi Zhang, Erin Cherry, Bobby Zhou, Huaishu Peng, and Tanzeem Choudhury. 2014. BodyBeat: a mobile system for sensing non-speech body sounds.. In *Proc. MobiSys*. ACM, 2–13.
[45] Ann Mette Rekkedal. 2012. Assistive hearing technologies among students with hearing impairment: Factors that promote satisfaction. *Journal of Deaf Studies and Deaf Education* 17, 4 (2012), 499–517.
[46] Android Studio. 2017. Battery Historian Charts. https://developer.android.com/studio/profile/battery-historian-charts.html. (2017).
[47] Denis Tomé, Luca Bondi, Luca Baroffio, Stefano Tubaro, Emanuele Plebani, and Danilo Pau. 2016. Reduced memory region based deep Convolutional Neural Network detection. In *Proc. ICCE-Berlin*. IEEE, 15–19.
[48] Emmanuel Vincent, Nancy Bertin, Rémi Gribonval, and Frédéric Bimbot. 2014. From blind to guided audio source separation: How models and side information can improve the separation of sound. *Signal Processing Magazine* 31, 3 (2014), 107–115.
[49] In-Chul Yoo and Dongsuk Yook. 2008. Automatic sound recognition for the hearing impaired. *Transactions on Consumer Electronics* (2008), 2029–2036.

A SURVEY QUESTIONS

In this appendix, we list the questions we used for the three surveys in our work, including the pre-design survey in Section 3, the user interface usage survey in Section 5.3 and the user study survey in Section 6.5.

A.1 Pre-design Survey

(1) What sound-awareness tools do you use daily, if any?
☐ Cochlear implants ☐ Hearing aids ☐ Flashing lights ☐ Vibration beds
☐ Others _____
(2) I feel comfortable to wear my hearing aids.
☐ Agree ☐ Neutral ☐ Disagree ☐ N/A
(3) I find the visual clues of ceiling lights installed in the classroom and the dormitory is difficult to remember.
☐ Agree ☐ Neutral ☐ Disagree ☐ N/A
(4) I feel uncomfortable with the vibration beds in the dormitory.
☐ Agree ☐ Neutral ☐ Disagree ☐ N/A
(5) I need a complimentary sound-awareness tool in addition to my hearing aid / cochlear implants.
☐ Agree ☐ Neutral ☐ Disagree ☐ N/A
(6) What complimentary sound-awareness tool do you want? Check all that apply.
☐ Head-mounted tools ☐ Ear-plugged tools ☐ Phone Apps ☐ Tablet Apps
☐ Wristband Apps ☐ Others _____
(7) Do you have a smartphone or will you have a smartphone soon?
☐ Yes ☐ No, but I will get one soon
☐ No, but I want one ☐ No, I don't want to use smartphones for now
(8) I want to use a smartphone App in addition to my hearing aids as double insurances.
☐ Agree ☐ Neutral ☐ Disagree ☐ N/A
(9) I want to use a smartphone App as a temporary relief without hearing aids when I am
☐ Studying alone ☐ Sleeping ☐ Relaxing ☐ N/A
☐ Others _____

All questions are originally in Mandarin and translated into English. All the three questionnaires require the participant to fill in the personal information of age and gender, which is omitted here.

(10) I want to use a smartphone App as a temporary relief without hearing aids in places of:
☐ Classroom ☐ Library ☐ Dormitory ☐ Restaurant
☐ Home ☐ Others _____

(11) What non-speech sounds do you care about? Check all that apply.
☐ Every sound ☐ Doorbells ☐ Knocking on the door ☐ Telephone ring
☐ People laughing ☐ People shouting ☐ People crying ☐ People coughing
☐ Dogs barking ☐ Fire alarm ☐ Smoke alarm ☐ Police alert
☐ Wake-up alarms ☐ Microwave oven ☐ Kettle boiling ☐ TV turning on/off
☐ Others _____

(12) How often do you miss an important acoustic event currently? Please specify (*e.g.* once a day) _____.

(13) How often can you tolerate false notifications of acoustic events? Please specify (*e.g.* once a day) _____.

(14) How often can you tolerate the missing of an acoustic event? Please specify (*e.g.* once a day) _____.

(15) How often can you tolerate the wrong classification of an acoustic event? Please specify (*e.g.* once a day) _____.

(16) How much delay can you tolerate before you get the notification of an acoustic event? Please specify (*e.g.* 10 seconds) _____.

(17) What aspects do you think are important in the sound-awareness tool? Check all that apply.
☐ Accurate ☐ Location-independent ☐ Responsive ☐ Energy-efficient
☐ User-friendly ☐ Others _____

(18) Are you willing to label and upload audio clips to help us improve the performance of the sound-awareness tool?
☐ Yes ☐ Neutral ☐ No

(19) Do you have other requirements for a sound-awareness tool? Please specify _____.

A.2 User Interface Survey

(1) UbiEar is easy to operate.
☐ Agree ☐ Neutral ☐ Disagree

(2) Which acoustic event(s) have you subscribed? Check all that apply.
☐ Doorbells ☐ Knocking on the door ☐ People crying
☐ People coughing ☐ Fire alarm ☐ Smoke alarm
☐ Police alert ☐ Kettle boiling ☐ Microwave oven warning

(3) Which reminder mechanism(s) do you use? Check all that apply.
☐ Graphic view ☐ Vibration ☐ Flashing

(4) The event reminder mechanisms are sufficient.
☐ Agree ☐ Neutral ☐ Disagree

(5) What other reminder mechanisms do you want? Please specify _____.

(6) Do you have other comments on the user interface of UbiEar? Please specify _____.

A.3 User Study Survey

(1) When using UbiEar, I am _____. Check all that apply.
☐ Hard-of-hearing with hearing aids ☐ Deaf with hearing aids
☐ Hard-of-hearing w/o hearing aids ☐ Deaf w/o hearing aids

(2) Score your satisfaction on the response delay for each events. [Score 0-3: unsatisfied; Score 4-6: neutral; Score 7-10: satisfied]
- ○ Doorbells [ ] ○ Knocking on the door [ ] ○ People crying [ ]
- ○ People coughing [ ] ○ Fire alarm [ ] ○ Smoke alarm [ ]
- ○ Police alert [ ] ○ Kettle boiling [ ] ○ Microwave oven warning [ ]

(3) Score your satisfaction on the false event detection for each event. [Score 0-3: unsatisfied; Score 4-6: neutral; Score 7-10: satisfied]
- ○ Doorbells [ ] ○ Knocking on the door [ ] ○ People crying [ ]
- ○ People coughing [ ] ○ Fire alarm [ ] ○ Smoke alarm [ ]
- ○ Police alert [ ] ○ Kettle boiling [ ] ○ Microwave oven warning [ ]

(4) Score your satisfaction on the missing rate of event detection for each event. [Score 0-3: unsatisfied; Score 4-6: neutral; Score 7-10: satisfied]
- ○ Doorbells [ ] ○ Knocking on the door [ ] ○ People crying [ ]
- ○ People coughing [ ] ○ Fire alarm [ ] ○ Smoke alarm [ ]
- ○ Police alert [ ] ○ Kettle boiling [ ] ○ Microwave oven warning [ ]

(5) Score your satisfaction on the recognition accuracy for each event. [Score 0-3: unsatisfied; Score 4-6: neutral; Score 7-10: satisfied]
- ○ Doorbells [ ] ○ Knocking on the door [ ] ○ People crying [ ]
- ○ People coughing [ ] ○ Fire alarm [ ] ○ Smoke alarm [ ]
- ○ Police alert [ ] ○ Kettle boiling [ ] ○ Microwave oven warning [ ]

(6) Do you have other comments on UbiEar? Please specify _____.

Received February 2017; revised April 2017; accepted May 2017