Knocker: Vibroacoustic-based Object Recognition with Smartphones

TAESIK GONG, School of Computing, KAIST, Republic of Korea
HYUNSUNG CHO, School of Computing, KAIST, Republic of Korea
BOWON LEE, Department of Electronic Engineering, Inha University, Republic of Korea
SUNG-JU LEE, School of Computing, KAIST, Republic of Korea

While smartphones have enriched our lives with diverse applications and functionalities, the user experience still often involves manual cumbersome inputs. To purchase a bottle of water for instance, a user must locate an e-commerce app, type the keyword for a search, select the right item from the list, and finally place an order. This process could be greatly simplified if the smartphone identifies the object of interest and automatically executes the user preferred actions for the object. We present *Knocker* that identifies the object when a user simply knocks on an object with a smartphone. The basic principle of Knocker is leveraging a unique set of responses generated from the knock. Knocker takes a multimodal sensing approach that utilizes microphones, accelerometers, and gyroscopes to capture the knock responses, and exploits machine learning to accurately identify objects. We also present 15 applications enabled by Knocker that showcase the novel interaction method between users and objects. Knocker uses only the built-in smartphone sensors and thus is fully deployable without specialized hardware or tags on either the objects or the smartphone. Our experiments with 23 objects show that Knocker achieves an accuracy of 98% in a controlled lab and 83% in the wild.

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

Additional Key Words and Phrases: Object recognition; Object interaction; Smartphone sensing; Machine learning; Multimodal sensing

ACM Reference Format:
Taesik Gong, Hyunsung Cho, Bowon Lee, and Sung-Ju Lee. 2019. Knocker: Vibroacoustic-based Object Recognition with Smartphones. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 82 (September 2019), 21 pages. https://doi.org/10.1145/3351240

1 INTRODUCTION

Heading towards a connected world, a smartphone plays an essential role as the interface between the physical objects and online services. For example, we use smartphones to look up how to operate a coffee machine, purchase water on e-commerce apps, and control IoT devices such as turning on a lamp with an app. Although available through a sequence of simple taps on smartphones, connecting the physical world and smartphone services yet includes a cumbersome process, especially when it is used repeatedly. When purchasing goods through e-commerce smartphone apps for instance, a user has to follow a series of manual procedures, i.e., unlocking the phone, finding and launching the right app, locating the desired product inside the app, and placing an order. Had the smartphone known the object of interest and the following routine of the user's desired

Authors' addresses: Taesik Gong, taesik.gong@kaist.ac.kr, School of Computing, KAIST, Republic of Korea; Hyunsung Cho, hyunsungcho@kaist.ac.kr, School of Computing, KAIST, Republic of Korea; Bowon Lee, bowon.lee@inha.ac.kr, Department of Electronic Engineering, Inha University, Republic of Korea; Sung-Ju Lee, profsj@kaist.ac.kr, School of Computing, KAIST, Republic of Korea.

2474-9567/2019/9-ART82 $15.00
https://doi.org/10.1145/3351240

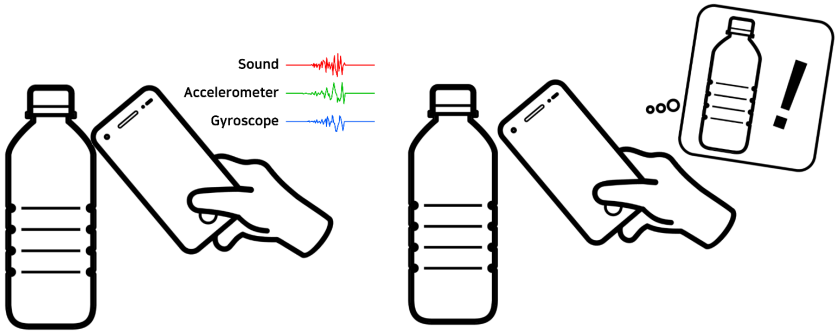

Fig. 1. An example knock on a bottle. Knocker identifies the object by analyzing a unique set of responses from the knock, and automatically launches a proper application or service.

action involving the object, it would have shortened the procedure and provided a more seamless and efficient interaction between the physical objects and smartphone services.

With the recent rise of speech recognition, voice command systems such as Apple Siri, Google Assistant, and Amazon Alexa have been suggested to provide quicker and easier interaction with the objects. Although the technology is promising, it still suffers from low accuracy [8] rooting from innate complexity of natural languages (e.g., numerous languages and dialects [37]), and privacy and security concerns [18, 30, 33, 42, 43]. Identifiable artificial markers are most commonly used to identify objects for interaction, such as barcodes, QR codes, and RFID tags [24, 39]. The largest problem in this, however, is that it requires every single object to be instrumented with a marker, which incurs additional cost in terms of both expense and efforts in deployment. Recently, a wide range of sensing approaches from visual [4, 6] to electromagnetic (EM) based sensing [22, 38, 40] have been studied to enable interaction with objects through object identification. However, existing methods have limitations. Visual sensing approaches are highly dependent on lighting conditions and alignment of the object in line of sight, or require a special hardware such as RGB-Depth cameras. EM sensing approaches require specialized EM sensors as well, and are applicable to only electrical appliances that emit EM signals. Their deployability is thus limited.

As a viable alternative, we introduce Knocker that identifies the object when a user simply "knocks" on an object with a smartphone. Figure 1 illustrates an example knock on a bottle. Knocker aims to identify a set of everyday objects that a user regularly interacts with and automatically launch a proper application or service. The basic principle of Knocker is leveraging a unique set of responses generated by the knock on an object according to its material, shape, size, etc. These responses are captured through a smartphone's built-in sensors: the sound from the microphone, and the motion from the accelerometer and gyroscope. With the multimodal sensor data, Knocker in turn performs object identification by applying Support Vector Machine (SVM) to classify the unique set of responses among others.

This new interaction method overcomes the limitations of existing work. Since Knocker leverages the unique response from the knock on an object, it requires no implementation on objects. Knocker does not limit its object coverage to electronic appliances. Furthermore, Knocker is fully functional on commodity off-the-shelf smartphones without any augmentation, as modern smartphones are equipped with microphone, accelerometer, and gyroscope. Moreover, its usability is not limited by lightning conditions (as in vision based approaches) or complexity of language processing (as in voice based approaches).

Utilizing accurate object identification from Knocker, we implement and present a wide range of applications, such as seamless online purchase and information retrieval about the identified object. We also demonstrate Knocker's distinct functionality, "multi-knock," which maps different services based on the number of knocks, to enable multi-function mapping to a single object. This functionality expands the input space for interaction.

To understand the performance of Knocker, we first evaluated it with 20 users and 23 everyday objects in a controlled lab environment. In addition to object identification accuracy, we measured the effectiveness of utilizing motion sensors, the impact of noise and underlying objects, the capability of distinguishing similar objects, and power consumption. We then took Knocker outside a quiet room and evaluated its in-the-wild real-time performance in terms of accuracy, latency, false positives and negatives through real-world experiments under diverse environments.

We make the following contributions: (i) we present a novel object identification method that leverages the unique set of responses generated from knocking on an object; (ii) we devise a sensing pipeline to accurately detect short knock signals among continuous sensor streams, which is robust to false positives and negatives; (iii) we implement 15 fully functional applications that demonstrate the applicability and usability of Knocker, especially its multi-knock functionality; and (iv) we performed an extensive evaluation of Knocker with 23 objects under various environments and conditions. Our results indicate that Knocker achieves around 98% accuracy in the lab and 83% accuracy in the wild with the identification latency of 229 ms.

2 RELATED WORK

We review different sensing approaches designed for object identification and interactive systems based on these approaches.

2.1 Acoustic-based Object Identification

Acoustic-based approaches leverage unique acoustic responses of each object in various ways [5, 11, 15, 16, 21, 23, 25, 26, 34]. ViBand [21] utilizes bio-acoustic signals composed of micro-vibrations that are generated by the objects and transmitted to the smartwatch accelerometers through the body. By modifying the kernel to increase the sampling rate of the accelerometer, ViBand achieves high accuracy in correctly classifying the objects on a commodity smartwatch. However, the set of detectable objects is restricted to mechanical or motor-powered devices that are designed to generate constant vibrations such as a blender, a coffee grinder, or a saw.

Impact-based approaches have been studied for extended application of acoustic sensing on non-mechanical everyday objects. Instead of utilizing the signals generated by the object itself, these approaches suggest exerting an impact on the object and analyzing the response signals for identification. Luo et al. [26] use the sound generated from knocking a marker pen to the object for object recognition, and Shi et al. [34] use only the sound of knocking with a knuckle wearing a smartwatch. In addition to sound, Knocker exploits the motion vibrations generated by a knock for higher accuracy and robustness to noise and environmental changes, which is demonstrated through our extensive evaluation.

Object identification using multimodal acoustic sensing was proposed in our previous study [11]. This work extends our previous work in the following ways: (i) we devise an improved sensing pipeline that is tolerant to external noises, (ii) we emphasize the user interface design and technology by suggesting and implementing various, fully functional applications, and (iii) we rigorously evaluate the system with a larger objects set under diverse conditions including in-the-wild experiments.

2.2 Other Object Identification Approaches

Traditional object identification systems such as barcodes, QR (Quick Response) codes, and other fiducial markers [16, 19, 31] attach artificial markers to each object. CyberCode [31] for example, uses a unique 2D barcode

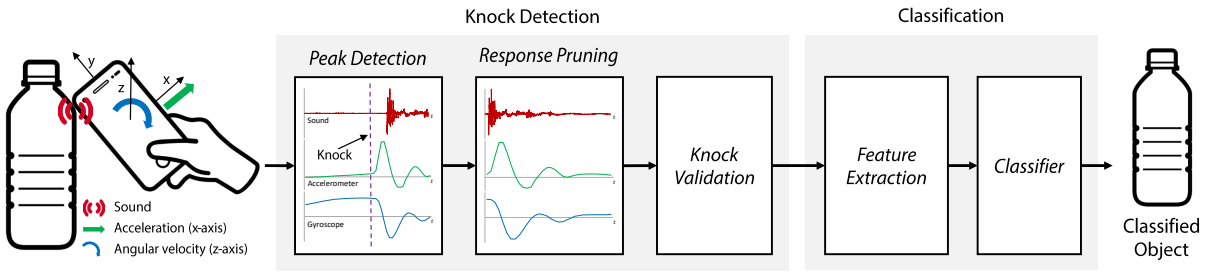

Fig. 2. Knocker system overview.

attached to each object as a visual tag to enable augmented reality applications. Embedding electronic RFID tags in objects has also been popular for recognizing objects and further providing interactions with the objects [24, 39]. Acoustic Barcodes [16] introduces identification tags with physical notches that produce a unique identifiable sound on the surface of an object. ObjectSkin [12] utilizes hydroprinted touch sensors and displays augmented on everyday objects. Although these approaches have high accuracy, it requires instrumenting every object of interest with a marker, which incurs a considerable amount of deployment effort and cost.

Vision-based solutions rely on the natural visual information of objects captured through the camera without per-object instrumentation [17, 29, 32]. For example, Maekawa et al. [29] recognized hand-held objects such as toothbrush using wrist-worn cameras. On top of image recognition, some work further utilized the phone's location and device orientation to distinguish similar-looking objects [4, 6]. Despite the popularity and capability to digest a diverse set of objects, the performance of visual approaches could degrade depending on lighting conditions, angle, and line of sight. Chen et al. [4] constructed a 3D model that is robust to lighting conditions and angle, but it requires an RGB-Depth camera that is seldom built in commodity smartphones.

Electromagnetic (EM) based approaches exploit sensing distinctive EM emissions of electrical, electro-mechanical, and electronic devices through EM sensors [22, 28, 38, 40, 41]. EM signals are natural byproducts in every electronic device. EM-based approaches thus do not require per-object instrumentation, but EM signals could only be sensed through a specialized sensor [41] or smartphones with additional hardware, such as antenna and microcontroller [40]. While leveraging distinctive EM signals from appliances is shown to be accurate, EM-based approaches are limited to identifying only electronic devices. Knocker instead targets a wider range of everyday objects including non-electronics, and it runs on commodity off-the-shelf smartphones.

3 KNOCKER

Knocker identifies objects by analyzing a unique set of response from the knock. We explain Knocker's design goals, basic principle, technical components and the multi-knock functionality.

3.1 Design Goals & Challenges

We present Knocker as an alternative object identification technique to mitigate the limitations of prior approaches. Specifically, we list our design goals and challenges as the following:

- **Handy interaction**: A desired object identification system should be ready for ease of use. It should simplify users' manual actions (such as executing a camera app to take a picture) required to trigger the desired service.
- **Real-time**: An object identification system should rapidly identify objects so that it would not incur high latency that would harm user experience. Therefore, the sensing pipeline and classification model should be designed carefully to support the real-time interaction.

- **Robustness**: Object identification should be robust to environmental changes such as external noises and the variety of underlying objects. Also, an object identification system should minimize unwanted triggers and undetected users' intentions, i.e., false positives/negatives.
- **Commodity-only**: The prevalent approaches in object identification augment objects with markers (e.g., QR-code and RFID) or require special-purpose hardware (e.g., RGB-D camera and EM sensors). We aim to be free of those augmentations or additional hardware so as to reduce the cost and effort and enable wide deployability.

3.2 Basic Principle

When a user knocks on an object with a smartphone (left illustration in Figure 2), the knock generates a unique set of responses based on the properties of the object, e.g., material, shape, and size. The basic principle of Knocker is analyzing the set of responses to identify each object. The most intuitive feature of the knock is the *sound* generated by the contact between the smartphone and the object, which is captured by the built-in microphone of the smartphone.

In addition to sound, a knock also exerts a force to the smartphone as the form of *acceleration* and *angular velocity*. Each object exhibits a different pattern of the force, and it is captured by the rapid changes in the built-in accelerometer and gyroscope sensor values in the smartphone. As using only sound is susceptible to noise, in addition to the knock sound, we leverage the accelerometer and gyroscope values that are both distinctive per object and noise-tolerant, to identify objects.

3.3 System Overview

Figure 2 illustrates the overview of Knocker. As Knocker aims to provide handy interaction without manual interventions from users, it continuously listens to sensors as a background process. When a user with a smartphone knocks on an object, a bottle in this example, Knocker detects the peak of the amplitudes of both the sound and the accelerometer values as the signal of a knock, and extracts only the knock-related segments from the raw data. Knocker then determines whether the current response is from an actual knock or falsely triggered from noise by analyzing the frequency distribution of the accelerometer values. Once the input is identified as a knock, Knocker calculates the features from the raw data. The features are put into a machine learning classifier and the classier outputs the classified object, e.g., a bottle.

3.4 Knock Detection

Knocker aims to support real-time identification on the knock while ensuring high accuracy with limited false positives/negatives. Manually launching the Knocker app (similar to launching a camera app in vision-based object identification) and providing an input (i.e., a knock) hinders our design goal of handy interaction. Knocker therefore continuously listens to sensor streams in the background in order to trigger a desirable service when a knock is detected. Since a "knock" response is very short (under 80 ms), Knocker must correctly detect and extract the right part of the knock response. Knocker thus involves a unique sensing pipeline, *Knock Detection*, to achieve the aforementioned goals.

3.4.1 Peak Detection. When a user knocks on an object, there appears an abrupt peak in the amplitude of both sound and accelerometer, and the values decrease as the time passes. Knocker utilizes this characteristic of knocks and monitors the built-in microphone and accelerometer in a smartphone to detect a knock. If there are peaks above the predefined thresholds in both sound and accelerometer, Knocker regards these peaks as a possible sign of a knock and begins buffering the knock-related streams of sensors (sound, accelerometer, and gyroscope values) for further computation.

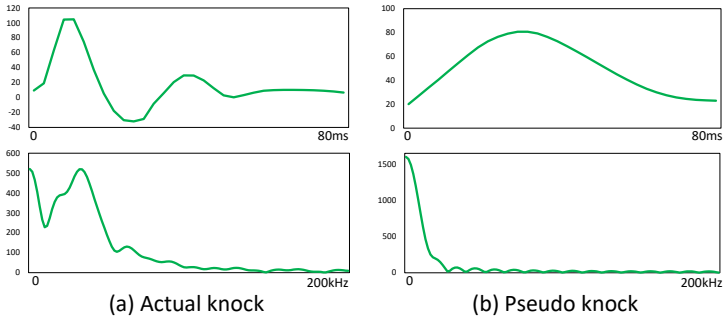

Fig. 3. Accelerometer response of (a) a knock and (b) a pseudo knock. The top figures represent the response in time domain and the bottom figures represent the response in frequency domain.

Because of the audio latency problem [7] mainly caused by the audio buffering process, there is delay between the peak of the accelerometer and that of the sound after a knock. Audio latency is difficult to predict and varies with the buffer size, smartphone models and OS versions [1, 10]. Since Knocker deals with a short signal of under 80 ms, the latency problem is critical in correctly extracting signal chunks of interest. With the intuition that knock responses start with peaks followed by oscillations, Knocker handle the audio latency by seeking a possible peak of sound after detecting the peak in the accelerometer. When Knocker observes the peak in sound, it regards the current sensor values as the response of a knock and proceeds to *Response Pruning*. The latency is around 20-150 ms in our prototype using Google Pixel 2 and the system disregards the peak of accelerometer unless a sound peak appears within 200 ms.

3.4.2 Response Pruning. From the series of raw data, only the knock-related responses must be extracted to exclude noise (e.g., ambient noise for sound and body movements for accelerometer and gyroscope) and minimize computation. This process, called response pruning, is started in Knocker by aligning the sound response with the accelerometer and the gyroscope. It utilizes the peaks in the aligning process and extracts 4,096 samples for sound and 32 samples for both accelerometer and gyroscope from the beginning of the knock. We selected these values based on our experiment study that investigates the duration of knock responses with 23 objects. We found that the duration varies per object, and ranges roughly from 20 ms to 80 ms. Given 48 kHz is the common sampling rate for the built-in microphone and 400 Hz for both accelerometer and gyroscope, this matches 85 ms for sound and 80 ms for accelerometer and gyroscope. This setting sufficiently captures the knock response and also minimizes containing data unrelated to a knock and thus computation overhead.

3.4.3 Knock Validation. The peak detection approach could be susceptible to accidental changes in both accelerometer values and sound. For example, a user who is swinging her smartphone in a noisy environment might trigger Knocker. This false positive could harm user experience. To reduce false positives, Knocker evaluates whether the current response is from an actual knock. We compared the responses from a real knock and a *pseudo* knock (swinging the phone in the air as if knocking on an object). Figure 3 shows the accelerometer responses from the actual knock and the pseudo knock are different; a real knock has more high-frequency components because of the rapid changes of the accelerometer values, while a pseudo knock has more low-frequency components. We define the ratio of the sum of the high-frequency components (higher than 15 Hz) to the sum of the low-frequency components (lower than 15 Hz) and use this ratio to examine whether the current streams of

Table 1. Features used in the classifier.

Raw sensor stream	Extracted Features
Sound (4,096)	Magnitude spectrum (2,049), Log magnitude spectrum (2,049), MFCCs (104)
Accelerometer (32)	Magnitude spectrum (129)
Gyroscope (32)	Magnitude spectrum (129)

inputs are a knock, as follows:

$$FSum_h = \sum_{f_i > 15\ Hz} f_i,$$
$$FSum_l = \sum_{f_i < 15\ Hz} f_i,$$
$$Ratio = FSum_h / FSum_l$$

where f_i is each element of the spectrum from the accelerometer response, $FSum_h$ is the sum of high-frequency components, and $FSum_l$ is the sum of low-frequency components. We regard a knock as valid if the ratio is greater than 2 in the experiment.

3.5 Classification

3.5.1 Feature Extraction. We use three types of features for sound: the magnitude spectrum, the log magnitude spectrum, and mel-frequency cepstral coefficients (MFCCs) [44]. In addition to the magnitude spectrum derived from the Fast Fourier Transform (FFT), we also use the log magnitude spectrum. While the magnitude spectrum effectively represents the prominent peaks in certain frequencies per object, the log magnitude spectrum boosts frequency contents of the acoustic signal with relatively low power that might also carry unique information per object. According to our experiments, using both sets of features gives higher classification accuracy than the individual ones.

The MFCCs, commonly used for automatic speech recognition, are widely used features derived by spacing the frequency bands according to the human auditory system [5, 9, 26, 35]. Our intuition behind using MFCCs is that human can discern the distinctive knock sounds from different objects.

We use the magnitude spectrum of the x-axis of the accelerometer and the z-axis of the gyroscope in consideration of the orientation of the knock with the length-256 FFT. To obtain higher frequency resolution, we apply zero-padding to the 32 samples of the gyroscope signal. These features are summarized in Table 1. The feature sets were determined after exploration of feature selection processes and different sets of features, such as statistical features (mean, variance, etc.), pair-wise band ratio, and derivatives. However, they degraded the accuracy or only slightly improved the accuracy at the expense of calculation overhead.

3.5.2 Classifier. We employ a sequential minimal optimization (SMO) based Support Vector Machine ($c = 1.0$, $\epsilon = 0.01$, polynomial kernel with $E = 1.0$) as the classifier, provided by the Weka machine learning toolkit [14]. SVM is a widely used machine learning technique that constructs an optimal hyperplane for classification. We adopt SVM since it requires less training data and runtime complexity compared with deep learning techniques and outperforms other classifiers in our experiments. Also, SVM models are well known for its resistance to overfitting due to high dimensionality owing to regularization. We tune the hyperparameters of regularization to minimize overfitting.

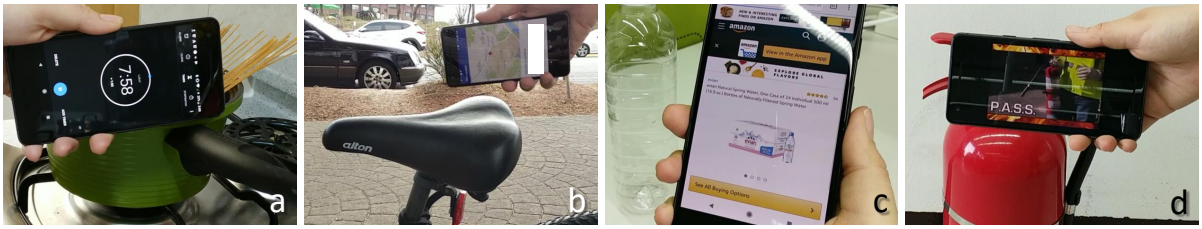

Fig. 4. Object-specific applications: knocking on (a) a pot to launch a cooking timer, (b) bicycle to mark the parking location, (c) a water bottle to order online, and (d) a fire extinguisher to view the usage instructions.

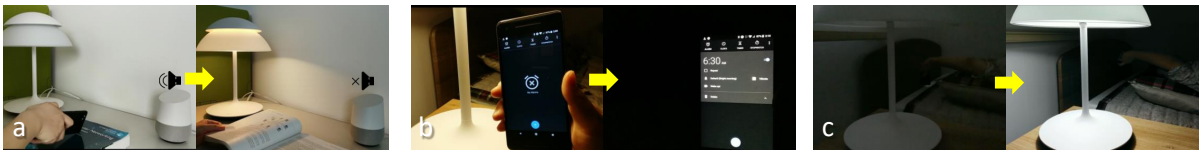

Fig. 5. Context-based device control: knocking on (a) a textbook to turn off the music and turn on the light for studying; (b) the headboard at night to turn off the light and set the alarm and the Do Not Disturb mode; and (c) the headboard in the morning to dismiss the alarm, turn on the light and listen to news.

3.6 Multi-Knock

A knock intrinsically has a discrete characteristic; in other words, one can count the number of knocks. We could leverage this discrete nature of a knock to expand the input space; one can map a different function to a different number of knocks in sequence. For instance, a single knock to a coffee machine launches the user manual, and a double knock links to the coffee bean purchase service. This concept is similar to the single and double clicks of computer mouse or touch systems. The capability of expanding the input space is a distinctive characteristic of Knocker compared with existing methods such as tag-based [16, 19, 31] and continuous sensing [21, 22, 28, 38, 40, 41] approaches. We implement multi-knock by counting the number of sequential knocks. Specifically, if the classified object is registered in the multiple-knock enabled object list, Knocker waits for up to 500 ms (empirically chosen but configurable) for additional knocks, counts the number of knocks, and maps to the predefined application according to the object and the number of knocks.

4 APPLICATIONS WITH KNOCKER

We introduce various example applications of Knocker. We implemented these applications that are fully functional on commodity off-the-shelf Android phones (please see the Supplemental Video). Our knock detection and classification pipeline runs as a background service. Once the object is identified, the service executes the predefined action through Android Intent, according to the classified object.

4.1 Object-Specific Applications

Knocker can trigger applications that are related to the use of the identified object. For example, using a cooking pot often involves the use of a timer when cooking time-sensitive food, e.g., pasta or eggs. When a user knocks on a pot handle, Knocker can automatically start the timer based on the user's preferences, e.g., ten minutes for spaghetti (Figure 4(a)). For a bicycle rider, Knocker can record the latest parking location when the user knocks on the saddle (Figure 4(b)). Knocking on consumer goods can launch an appropriate e-commerce app or

Fig. 6. Multi-knock applications: knocking on (a) a laptop once to transmit a photo from the smartphone, and twice to receive an audio file from the laptop; (b) a coffee machine once to open up the manual, twice to purchase coffee beans online, and three times to call the service center; and (c) a guitar once to open a tuner, twice to open a metronome, and three times to show sheet music.

website and retrieves the result of searching the object (Figure 4(c)). People often interact with shared objects that require a piece of information (e.g., instructions, password, etc.) to use. Knocker can shorten the information retrieval process with a knock. Especially for emergency situations such as fire, it can provide quick usage instructions (Figure 4(d)).

4.2 Context-Aware IoT Control

Knocker can infer the user's context by identifying the object in interaction and control the context-related IoT devices. Knocking on a book for instance, could be considered as "going to study" (Figure 5(a)). Knocker can set up a studying environment by turning on the smart lamp and muting the music.

Adding more contextual feature can enhance Knocker's functionality. For example, utilizing time information can make Knocker provide different actions depending on the time of day. Knocking on the headboard at night infers going to bed, thus Knocker sets the alarm, activates the Do Not Disturb mode, and turns off the lights (Figure 5(b)). On the other hand, knocking on it in the morning infers waking up, and Knocker dismisses the alarm, turns the lights on, and triggers headlines briefing in voice (Figure 5(c)).

4.3 Multi-Knock Applications

For certain objects, users interact with them in various ways. Knocker takes advantage of the discreteness of the knock to support multiple applications for an object through "multi-knock." Figure 6 illustrates three objects that leverage the multi-knock feature. If a user knocks once on a laptop, it transmits the latest photo taken from the smartphone to the laptop, while knocking twice downloads a music file from the laptop to the smartphone (Figure 6(a)). For a coffee machine, knocking once opens the user manual, twice links to the coffee bean or capsule seller, and three times calls the customer service center (Figure 6(b)). For a guitar, knocking once launches a guitar tuner, twice launches a metronome, and three times shows sheet music (Figure 6(c)).

5 IN-LAB EVALUATION

The goal of our in-lab study is to evaluate the performance of Knocker with different objects and users. We tested Knocker to answer the following questions: (i) How accurate is Knocker with different knock styles of different users? (ii) Can Knocker leverage training data of others without personalized training data? (iii) How do environmental changes affect Knocker's accuracy? (iv) Can Knocker distinguish similar objects? (v) How much energy does Knocker consume?

5.1 Experiment Settings & Procedures

We conducted an IRB-approved user study experiment with 20 participants (aged 20-40, mean 24.4; 5 females; 1 left-handed). The 23 objects used in the experiment are shown in Figure 7. To consider a real-world situation, each object was knocked on while placed in its typical use condition; for instance, the smartwatch was worn on a wrist, a guitar was on a knee, a hair dryer was grabbed by the hand, etc. Each object has a knock area (dotted

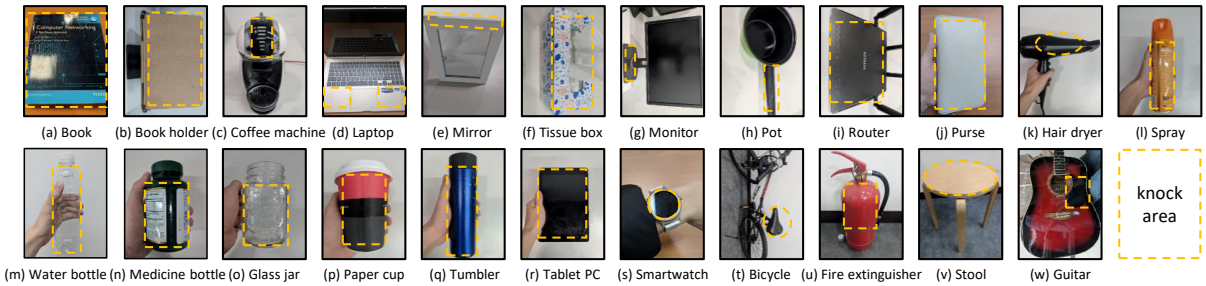

Fig. 7. 23 objects used in our evaluation. Each object is placed on its usual condition (e.g., laptop on a desk, smartwatch on a wrist, guitar on knees) and is knocked on the knock area

Table 2. Average identification accuracy with per-user and leave-one-user-out classifiers tested in a quiet room. Accuracy of individual feature set is specified. "Sound only" refers to using MagSpec, LogSpec, and MFCC, while "All" refers to using all features.

	MagSpec	LogSpec	MFCC	Acc.	Gyro.	Sound only	All
Per-User Classifier	93.26%	91.30%	96.74%	48.04%	37.17%	98.26%	**98.26%**
Leave-One-User-Out Classifier	90.65%	88.70%	92.83%	43.26%	29.78%	95.00%	**96.74%**

rectangles in Figure 7) that we consider is likely to be knocked when in practical use; for example when cooking, it would be natural to knock on the grip of the pot, not the body. Different users might prefer to knock different areas and we discuss the knock area in Section 7.1.

We used Pixel 2 smartphones for experiments. Two of the authors performed a pilot study with four different phone models (Pixel 2, Pixel 1, Essential Phone, and Nexus 5) and their accuracy variation was within 5%. We chose Pixel 2 that showed the best performance among them.

Training dataset was collected in a quiet room. Participants were asked to knock on various parts of the objects within the knock area, in order to prevent over-fitting to a specific spot of the object. We put no constraints on the strength and speed of the knock, the hand and body postures, and their smartphone grip. To avoid over-fitting to a specific smartphone grip and hand/body postures, (i) we divided training data collection into 10 rounds; (ii) participants knocked every 23 objects 10 times in each round; and (iii) participants regripped the phone every round. For the smartwatch, users took it off and put it back on every round. In total, each user collected 2,300 knocks for training data. One knock per object was collected by each participant for the test set.

5.2 Results & Analysis

We first analyze the classification accuracy of Knocker by comparing per-user classifiers and leave-one-user-out classifiers, and then visualize knock responses to further understand the performance of Knocker.

5.2.1 Per-User Classifier. As Knocker targets everyday objects that a person regularly interacts with, we first evaluate Knocker's performance using per-user classifiers. For each user, we used 100 knocks to each object for training (a total of 2,300 for training) and each user tested each object once after training. The result is shown in the first row of Table 2. The average accuracy across 20 users is 98.26% (SD=3.48%). The variation in accuracy might be caused by different knock styles from different users (e.g., grip, strength, etc.).

For per-user classifiers, data collection process might be burdensome for individuals. We thus analyze the accuracy based on the amount of training data per object (see Figure 8). As expected, accuracy improves with

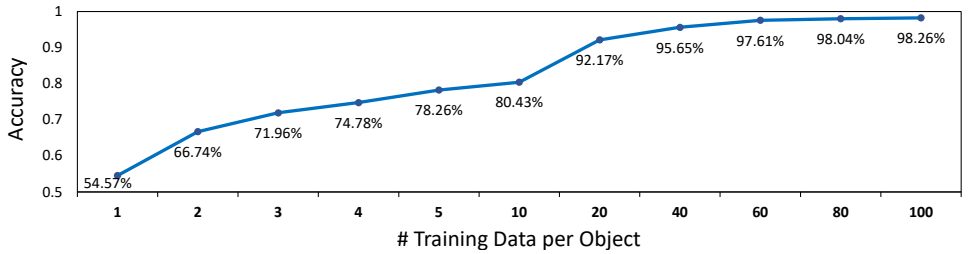

Fig. 8. Accuracy as a function of the amount of training data per object. Per-user classifiers are used.

	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)	(o)	(p)	(q)	(r)	(s)	(t)	(u)	(v)	(w)
Book (a)	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Book holder (b)	0	19	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Coffee machine (c)	0	2	17	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Laptop (d)	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mirror (e)	0	1	0	0	18	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tissue box (f)	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Monitor (g)	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Pot (h)	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Router (i)	0	0	0	1	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Purse (j)	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0
Hair dryer (k)	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0
Spray (l)	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	1	1	0	0	0	0	0
Water bottle (m)	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0
Medicine bottle (n)	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0
Glass jar (o)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0
Paper cup (p)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0
Tumbler (q)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0
Tablet PC (r)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0
Smartwatch (s)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	19	0	0	0	0
Bicycle (t)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0
Fire extinguisher (u)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	19	0	0
Stool (v)	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	17	0
Guitar (w)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20

Fig. 9. Confusion matrix of 23 objects. Rows are actual classes and columns are predicted classes. Leave-one-user-out (LOUO) classifiers are used.

more training data. Interestingly, only ten knocks per object achieves around 80% accuracy with 23 objects. With 60 knocks per object, Knocker exceeds 97% accuracy. In addition, we investigate the time taken to collect the training data. For each user, we logged the time taken to finish collecting 2,300 knocks. On average, 39.8 minutes were taken (SD=9.2 minutes) for data collection. This is roughly less than 2 minutes for collecting 100 knocks per object. We believe this is not a huge burden for a user to scale Knocker to possible new objects of interest.

5.2.2 Leave-One-User-Out Classifier. Whether Knocker can be trained with existing data from others is an important question to build a general model of Knocker. There could be a concern that the knock responses are overfitted to the personal training data. To evaluate whether utilizing training data from others would still provide high accuracy, we used the model trained from the data collected by the other 19 users and tested on

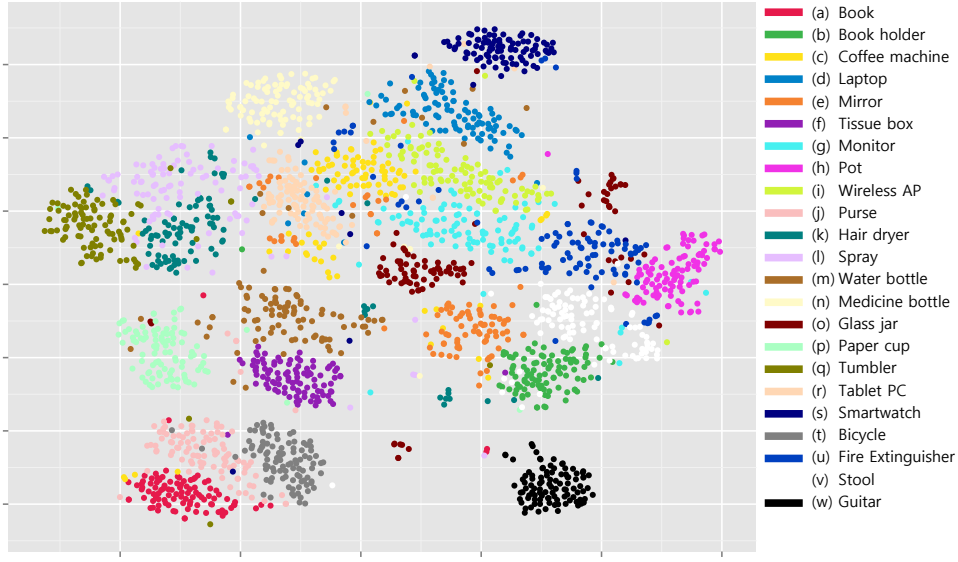

Fig. 10. T-SNE visualization of knock responses. Randomly selected 100 samples per object from the 20 subjects are used (iteration=5K, perplexity=40).

each user, i.e., leave-one-user-out. The average accuracy of this leave-one-user-out (LOUO) classifier (Table 2) is 96.74% (SD=4.10%), which is comparable to the accuracy when using the individual training data (around 1.5% degradation for the "All" case).

This result indicates that while personal dependency exists to certain extent, the knock responses are generalizable without individual training data. This result is also in line with the possibility of using others' data. Utilizing the already collected data by others would significantly reduce the users' burden on data collection. This could especially be useful for a group of users that share the specific model of an object (e.g., a coffee machine). Building the classification model using crowdsourced knock data can facilitate deployment at scale with low training overhead [2, 3, 13, 20]. For the rest of evaluations, we focus on the accuracy of LOUO classifiers to remove overfitting to each user and evaluate the generalizability of Knocker.

Figure 9 shows the confusion matrix for all 23 objects with LOUO classifiers. Since we tested each object once per user, the sum of each row is 20. The darker the cell, the higher the number. As expected, objects with similar characteristics were sometimes misclassified as each other (e.g., (l) spray and (q) tumbler). Overall, Knocker has high accuracy among all objects without being heavily biased to specific objects.

5.2.3 Visualization of Knock Response. To further understand how the knock responses are distinguishable between objects, we leverage t-distributed Stochastic Neighbor Embedding (t-SNE) [27] to visualize knock responses of different objects. T-SNE is a widely used dimensionality reduction technique for visualizing data from a high-dimensional space to low in a way that similar points are plotted nearby. Figure 10 illustrates t-SNE for the 23 objects, plotted with randomly selected 100 samples per object from all 20 subjects (iteration=5K, perplexity=40).

Figure 10 intuitively shows similar objects are close to each other (e.g., (b) book holder and (v) stool, which are both made of woods), while dissimilar objects are far from each other (e.g., (a) book and (b) book holder). Although similar objects are adjacent to each other, intra-object points are more close to each other than inter-object

Table 3. Average identification accuracy with different noise types. Noise levels are specified in dBA. Quiet is the same with the leave-one-user-out case in Table 2.

	Noise	MagSpec	LogSpec	MFCC	Acc.	Gyro.	Sound only	All
Quiet (LOUO)	30-35 dBA	90.65%	88.70%	92.83%	43.26%	29.78%	95.00%	**96.74%**
CNN (LOUO)	50-65 dBA	92.17%	89.57%	86.09%	41.96%	27.61%	93.91%	**95.43%**
Music (LOUO)	65-72 dBA	88.04%	83.48%	66.74%	39.78%	27.17%	88.26%	**90.22%**

Fig. 11. Six underlying objects used in the experiment.

points, which enables Knocker to identify objects. The fact that inter-object variability is higher than intra-object variability when collected from different subjects validates the generalizability of Knocker across users.

5.3 Impact of Environmental Changes

There are various noises that could degrade the performance of sound-based systems. Moreover, objects could have different underlying surfaces (e.g., a laptop on a table or a lap) that might affect the knock responses. We thus examined the effect of environmental changes on Knocker by noise types and underlying objects. We also measured how much motion sensors improve the accuracy when sound features suffer from noise. Note that we used the same LOUO classifiers trained with the 23 objects only in the quiet room.

5.3.1 Noise. We evaluated Knocker on two different noises: (i) with CNN news turned on (CNN), (ii) with pop music turned on (Music). We played CNN and music respectively from another smartphone placed one meter away from 23 objects with 100% volume, in the same office room. We measured the noise levels from each environment with a decibel meter. For both environments, each participant knocked on each object once for testing.

Table 3 reports the average accuracy with 20 users in different environments. We specify the accuracy with each feature set in order to investigate the contribution of each feature. We observe that the louder the noise, the worse the result. We found turning on CNN had little impact on accuracy. We attribute this result to the dominance of the knock sound over the ambient noise, due to the short distance from the knock to the built-in smartphone microphone.

5.3.2 Underlying objects. Objects might be placed on different objects underneath when moved. For instance, a laptop could be on a desk or on a lap. We investigated whether the trained model on a one specific underlying object still remains accurate on such different underlying objects. Figure 11 shows the pictures of six underlying objects we tested. We determined the six underlying objects considering the thickness and size (a table vs. a desk), hard and soft materials (tables vs. mattresses), and object-specific condition (e.g., laptops on a lap, books on a book holder, etc.). We tested a subset of objects that are likely to be on each underlying objects using the LOUO classifiers trained with 23 objects on each condition described in Figure 7. Specifically, (a) book, (d) laptop, (j) purse, and (r) tablet PC were tested on Table, Desk, and Mattress; (a) book, (j) purse, and (r) tablet PC on Hand;

Table 4. Average identification accuracy with six different underlying objects. Tested objects used for each condition are specified.

	MagSpec	LogSpec	MFCC	Acc.	Gyro.	Sound only	All
Table (a,d,j,r; LOUO)	77.50%	85.00%	71.25%	33.75%	41.25%	87.50%	**93.75%**
Desk (a,d,j,r; LOUO)	67.50%	80.00%	76.25%	51.25%	50.00%	76.25%	**83.75%**
Mattress (a,d,j,r; LOUO)	62.50%	61.25%	55.00%	12.50%	12.50%	68.75%	**77.50%**
Hand (a,j,r; LOUO)	65.00%	68.33%	66.67%	61.67%	43.33%	68.33%	**80.00%**
Lap (d; LOUO)	55.00%	50.00%	50.00%	20.00%	0.00%	70.00%	**85.00%**
Book holder (a; LOUO)	40.00%	25.00%	70.00%	65.00%	70.00%	35.00%	**80.00%**

Fig. 12. Four similar object groups: three aluminum cans, three plastic bottles, three hardcover books and three softcover books.

(d) laptop on Lap; and (a) book on Book holder. Similarly, each participant knocked each object once for testing under each condition.

Table 4 shows the average accuracy with varying underlying objects. As expected, accuracy is affected by underlying objects and shows a larger gap when tested in a more heterogeneous condition compared to the trained condition (e.g., laptop on the table vs. on lap) Still, Knocker achieves around 80% accuracy on the six underlying objects showing the response could be discernible under the changes of underlying objects. Specifically, although accelerometer or gyroscope solely show relatively low accuracy, when they are combined with sound features they actually help classification showing over 10% improvement in three cases in Table 4.

We discovered from our experiments that multimodal sensing approach of Knocker proved its worth when the sound features on their own could not provide high accuracy. Nevertheless, testing under different conditions from the training environment degrades the performance especially under heavy noise or underlying objects having dissimilar matter.

5.4 Distinguishing Similar Objects

The intuition of Knocker that leverages the responses from the knock to identify objects, is that the object composition (e.g., material, shape, etc.) is different among objects of interest. We now investigate to what extent objects are distinguishable. We chose four groups of similar objects that have very similar characteristics. Figure 12 shows 12 objects we tested, which divided into four similar groups: aluminum cans, plastic bottles, hardcover books and softcover books. The data collection and the test methods were the same to those of in Section 5.1 (100 knocks for training, one knock for testing in a quiet room).

The average accuracy is 92.08% for the per-user classifier and 93.33% for the leave-one-user-out classifier (Table 5). Figure 13 shows the confusion matrix for the similar objects with LOUO classifiers. As expected, similar objects were sometimes mistaken for each other; for instance, cans were classified as other cans with higher probability than as bottles. Interestingly, Knocker shows high accuracy even among similar objects except for

Table 5. Average identification accuracy for the 12 similar objects.

	MagSpec	LogSpec	MFCC	Acc.	Gyro.	Sound only	All
Similar objects (Per-user)	89.58%	73.33%	85.00%	56.25%	40.42%	90.83%	**92.08%**
Similar objects (LOUO)	85.00%	77.92%	75.83%	46.25%	30.42%	90.42%	**93.33%**

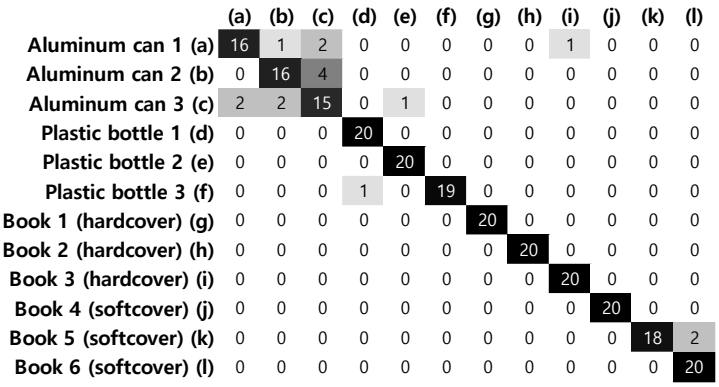

Fig. 13. Confusion matrix of similar objects with leave-one-user-out classifiers.

cans. Cans have the same material (aluminum), shape (cylinder), and content type (liquid). Cans could be in different sizes, but size alone is not a distinctive feature for accurate classification. While Knocker shows high accuracy among different everyday objects, it would not distinguish almost same objects with same material, shape, content type and size. For other similar objects, the combination of even slightly different characteristics of the objects make them distinguishable.

5.5 Power Consumption

As Knocker leverages multimodal sensing for the detection of knocks, one concern could be battery drain from the continuous reading of IMU sensors and microphone. We analyzed the power consumption of Knocker with Google Battery Historian [36] using the Pixel 2 prototype (described in Section 6.1). We performed a one-hour measurement of the battery percentage consumed for each of the two states: (1) *Knocker-on*: when Knocker is actively used and (2) *Knocker-off*: when Knocker is turned off with all other conditions being equal. For the *Knocker-on* state, one of the authors knocked the objects at the rate of one knock per minute. To isolate the amount of battery consumed by Knocker, we subtracted the measurement in the *Knocker-off* state from the value in the *Knocker-on* state. We did not include the service launching functionality for this test because there exist a wide array of services, and the energy consumption fluctuates depending on the service and its implementation.

The estimated battery percentage consumed by Knocker was 5.33%. Given the battery capacity of 2,700 mAh for Pixel 2, this is around 143 mAh per hour. For comparison, the battery consumption caused by the screen with the minimum brightness was 5.59% (150 mAh) for one hour. The main cause of battery drain is the continuous sampling of IMU and microphone. This could be mitigated when combined with an activation feature, such as when the phone is unlocked, or detection of the users' grip. We report users' feedback on the battery consumption due to the continuous sensing in Section 7.2.

Table 6. Average identification accuracy in the real-world experiment.

	MagSpec	LogSpec	MFCC	Acc.	Gyro.	Sound only	All
Real-world experiment	62.84%	65.80%	71.12%	39.70%	25.16%	77.08%	**83.02%**

6 REAL-WORLD EXPERIMENT

We took Knocker outside the lab environment and evaluated its performance under uncontrolled real world environments. We measure (i) real-world accuracy of Knocker, (ii) classification latency, and (iii) false positives/negatives.

6.1 Settings

We used a Pixel 2 phone using Weka machine learning toolkit [14] for implementing the Knocker online classifier. We trained the model with the training data collected under a quiet room with the 20 users for the 23 objects from the in-lab study (Section 5.1). We recruited 10 participants (6 females; aged 20–32) and deployed the phone for them to conduct the real-world experiment. We distributed four or five objects among the 23 objects to the participants (for their convenience of carrying objects considering the volume and weight) and each object was tested by two participants. Each participant tested the objects with the prototype in five different environments in diverse indoor and outdoor environments based on their daily routine. The environments included residential areas (house kitchen, living room, dormitory room, dormitory lounge, etc.), study/work places (seminar and lecture rooms, library, auditorium, corridor, lobby, elevator, parking lot, etc.), eating places (cafe, snack bar, restaurants, etc.), sports and cultural activity areas (gym, indoor basketball court, outdoor soccer field, lakeside, Roman theatre, instrumental practice room, etc.), transportation (in a bus or taxi, street, and bus stops), etc. These environmental changes naturally involve factors that could affect the performance of Knocker such as noise types (people speaking, music, vehicles, etc.), underlying objects (desk, floor, chair, hand, etc.), echoic characteristics (room, outdoor, etc.), and so on. We ensured that the environments were different from the data collection site and there were no identical environments. The users tested the objects five times per environment. Note that we did not give a specific guideline for knocking except for the knock area, and thus the participants knocked naturally with their preference (e.g., knock speed and strength). To prevent performing similar knocks, there was an intentional break between knocks and the users did not test the same object in a row. In total, we evaluated 1,150 real-world knocks (50 per object) from 10 users in 50 different environments.

6.2 Results

6.2.1 Accuracy. Table 6 shows the accuracy for the real-world experiment among ten users. Since there were various factors that degrade the performances of Knocker in diverse environments, the accuracy of Knocker in the real-world (83.02%) was lower than in the quiet room (96.74% for LOUO in Table 2). Note that while accelerometer and gyroscope features by themselves could not achieve high accuracy, they boosted the accuracy of around 6% when they revwere combined with the sound features.

Figure 14 shows the confusion matrix for the real-world experiment. We found that smaller objects that are often used on a surface are susceptible to the underlying object changes (e.g., book, mirror, and router) as the sound features generated from the objects are screened by the sound generated from the underlying object. But we believe users would not often change the underlying surface for the objects, as we usually put these objects in a certain fixed position. For these objects, additional training data for the position would improve the performance at a similar rate to Figure 8.

	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)	(o)	(p)	(q)	(r)	(s)	(t)	(u)	(v)	(w)
Book (a)	25	1	0	0	0	5	0	0	0	16	0	1	2	0	0	0	0	0	0	0	0	0	0
Book holder (b)	0	46	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Coffee machine (c)	0	0	33	0	4	0	1	0	0	1	0	3	4	0	0	0	1	0	0	2	1	0	0
Laptop (d)	0	0	1	39	0	0	1	0	0	4	0	0	3	0	0	0	0	2	0	0	0	0	0
Mirror (e)	0	1	11	0	19	0	6	0	0	0	2	3	0	1	0	0	3	1	0	0	0	3	0
Tissue box (f)	0	0	0	0	0	47	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
Monitor (g)	2	1	0	0	5	0	40	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
Pot (h)	0	0	0	0	2	0	7	41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Router (i)	0	0	7	3	1	0	8	0	29	0	0	0	1	0	0	0	0	0	0	0	0	0	1
Purse (j)	2	0	0	0	0	1	0	0	0	45	0	0	1	0	0	0	0	0	0	1	0	0	0
Hair dryer (k)	0	1	0	0	0	0	0	0	0	0	48	0	0	0	0	0	1	0	0	0	0	0	0
Spray (l)	0	0	0	0	1	1	0	0	0	0	1	46	0	0	0	0	0	0	0	0	0	0	1
Water bottle (m)	0	0	0	0	0	9	0	0	0	0	0	0	40	0	0	0	0	0	0	1	0	0	0
Medicine bottle (n)	0	1	0	0	0	0	2	0	0	0	0	0	0	47	0	0	0	0	0	0	0	0	0
Glass jar (o)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	49	0	0	0	0	0	1	0	0
Paper cup (p)	2	0	0	0	3	6	0	0	0	1	0	0	4	1	0	33	0	0	0	0	0	0	0
Tumbler (q)	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	48	0	0	0	0	0	0
Tablet PC (r)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0
Smartwatch (s)	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	2	0	0	46	0	0	0	0
Bicycle (t)	1	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	45	0	0	0
Fire extinguisher (u)	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	46	0	0
Stool (v)	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	44	0
Guitar (w)	0	3	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	43

Fig. 14. Confusion matrix of 23 objects for the real-world experiment. Rows are actual classes and columns are predicted classes.

6.2.2 Latency. We evaluated the time taken from the knock to the completion of the classification on the online classifier. We logged the time taken for each step of the Knocker's pipeline and averaged over all knocks in the real-world experiment. It took 140 ms for buffering the sensor values and validating the knock, 37 ms for feature extraction, and 50 ms for the classification. To sum up, the end-to-end latency was around 229 ms.

After the real-world experiment, we conducted a post user survey to further understand whether Knocker provides real-time interaction. We asked the participants "Does Knocker provide *real-time* interaction service?" and got answers in Likert scale (1: Strongly disagree; 5: Strongly agree). All participants reported no perceived delay with the average score of 5 points. Some participants mentioned "I did not perceive any delay at all." and "it is a lot faster than manually looking for apps and services." We believe around 229 ms delay does not hinder user's experience for real-time interaction with objects through Knocker. The nearly perceived delay of Knocker offers the advantage of quick interaction compared to existing approaches such as speech input for which users have to go through the at least a few seconds sequence of saying "Hey Google/Siri", waiting and checking if their assistant is awaken, and speaking the full-sentence command.

6.2.3 False Positives & Negatives. Minimizing false positives and negatives is important for the usability of Knocker. Knocker could falsely trigger (false positive), for example when a user makes hand gestures with the smartphone in hand in a noisy setting. This unwanted activation could harm the user experience. User experience with Knocker would also be unpleasant if it is not triggered when a user knocks on an object (false negative).

We recorded false positives and negatives during the real-world experiment where natural activities were involved such as different physical activities (walking, taking stairs, and riding a bicycle, bus, and taxi), different phone locations (in hand, inside a pocket, or in a bag), actions of putting down the phone, etc. There was one

false positive from one user out of the 10 participants during a total of 386 minutes experiment. This result shows that Knocker works well under various everyday activities and our knock validation algorithm is effective. There were 33 false negatives among 1,150 trials (2.8%). This was largely due to different knock strengths from different users as Knocker uses predefined thresholds for detecting knocks, which is configurable.

7 DISCUSSION

We discuss limitations of Knocker and suggest future work.

7.1 Knock Area

We assumed in our experiments that each object has an area to be knocked with practical considerations (e.g., the grip for a pot). This assumption requires less training data and simplifies the model compared with using the whole body of objects as the knock area, as different part of an object could generate different responses from the knocks (e.g., the body versus the cap of a water bottle). However, users might prefer to knock different areas.

We performed another experiment with 20 participants to understand where the users prefer to knock and whether they are different from our predefined areas. After a brief explanation of how Knocker works to each participant, we asked them to knock the 23 objects with the instruction to *"knock this object as if you initiate the interaction with the object"* and compared where they knocked on each object. Among all trials, 68% (315/460) matched our predefined knock areas. We also measured the degree of knock area variability among users by recording the unique areas knocked by users. The median value of the number of knocked areas for objects is 2 (min: 1 in the glass jar; max: 5 in the bicycle and guitar). This result shows the variability of knock areas is limited, and it is possible to use the entire body as the knock area by collecting training data from each part of the objects. Moreover, we could leverage the different responses from the same object for a new interesting application: mapping different applications for each part of an object.

7.2 Always Listening

As Knocker continuously listens for the sensor inputs, there could be privacy and energy concerns for the always-listening system. After the real-world experiment (Section 6), we asked participants how they thought about the device always listening. Prior to asking questions, we briefly explained Knocker's working mechanism of continuously listening with microphone, accelerometer, and gyroscope. We asked two possible user concerns regarding continuous listening: privacy and energy consumption. After obtaining their opinion, we also asked their willingness to use Knocker despite the concerns in the Likert scale (1-5).

Participants' willingness was 4.00 in the average Likert point (SD=0.67). The priority between privacy and energy consumption varied by participants, and some did not care about either issues considering the usefulness of Knocker. People who were less concerned with battery drain were those who "have the phone always charged." Regarding privacy, participants were fine as long as the microphone recordings are not saved or transferred to the server. Some participants noted that the always-listening mechanism is similar to the existing voice control systems. Several participants mentioned if there is an on-off feature for Knocker, both concerns would be mitigated.

7.3 Content Amount

Since Knocker identifies objects by analyzing distinctive knock response from the objects, it could be challenging to accurately identify objects that could have variation in themselves, such as changes of the amount of content. We observed that knock responses are slightly different when the content amount varies (e.g., the amount of pills in a medicine bottle changes). For these objects, the current prototype of Knocker would fail to consistently identify them correctly. Similarly to Snap-To-It [6] that updates the classifier with newly collected data, gradually

improving the model could enable identifying such gradually changing objects. Predicting the content amount by analyzing the different response could be an interesting future work.

7.4 Damaging the Phone

One might be concerned that knocking on objects could damage the smartphone (and the objects). In our experiments, no smartphone was damaged. We believe the manufacturers make smartphones durable, even withstanding a free-fall test. Nevertheless, covering the smartphone with a case would ease such concern. Two of the authors performed a pilot study where we trained and tested with a smartphone covered in a slim rubber case in the quiet room setting, and the averaged accuracy was 99.1%.

7.5 User Experience Study

In this paper, we focus on the technical aspect of object identification with a commodity smartphone and suggest possible applications of Knocker. As future work, studying the user interaction occurring after identification would add valuable insights on how users actually utilize Knocker in daily life. For example, it would be interesting to discover for what objects and functionalities people prefer to use Knocker over other methods; whether users want to distinguish two object instances of the same model; and how users feel about the training process. For a specific group of users such as visually impaired people, Knocker could open a new possibility to retrieve information from and interact with objects, since vision-based methods suffer from the camera alignment issue [17].

8 CONCLUSION

Knocker is a novel object identification system that only requires a smartphone knock. It does not rely on object augmentation or specialized hardware. Empowered with smartphone sensor fusion and machine learning, it achieved 98% accuracy in a quiet room when tested with 23 objects. We demonstrated the uniqueness of knock responses among different users through leave-one-user-out classifiers.

When the Knocker prototype was further tested in the real world where various types of noise and underlying objects changes were naturally involved, it still provided a reasonable accuracy of 83% showing the effectiveness of multimodal sensing with a latency of 229 ms. As our fully functional applications demonstrate, Knocker enables new convenient interaction method with everyday objects without having to manually provide input to the smartphone. Since Knocker can be used by people who have access to smartphones and can knock, we expect it could be an easily and widely deployable system for object identification and interactions.

ACKNOWLEDGMENTS

This work was supported in part by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2017M3C4A7083534), Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.B0717-16-0034, Versatile Network System Architecture for Multi-dimensional Diversity), the Industrial Technology Innovation Program funded by the Ministry of Trade, Industry & Energy (10073154), and the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2018S1A5A2A03037308).

REFERENCES

[1] Android. 2018. Audio Latency Measurements. Retrieved September 20, 2018 from https://source.android.com/devices/audio/latency_measurements.

[2] Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, and Tom Yeh. 2010. VizWiz: Nearly Real-time Answers to Visual Questions. In *Proceedings of the 23Nd Annual*

ACM Symposium on User Interface Software and Technology (UIST '10). ACM, New York, NY, USA, 333–342. https://doi.org/10.1145/1866029.1866080

[3] Joseph Chee Chang, Saleema Amershi, and Ece Kamar. 2017. Revolt: Collaborative Crowdsourcing for Labeling Machine Learning Datasets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 2334–2346. https://doi.org/10.1145/3025453.3026044

[4] Kaifei Chen, Jonathan Fürst, John Kolb, Hyung-Sin Kim, Xin Jin, David E. Culler, and Randy H. Katz. 2018. SnapLink: Fast and Accurate Vision-Based Appliance Control in Large Commercial Buildings. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 129 (Jan. 2018), 27 pages. https://doi.org/10.1145/3161173

[5] Sauvik Das, Gierad Laput, Chris Harrison, and Jason I. Hong. 2017. Thumprint: Socially-Inclusive Local Group Authentication Through Shared Secret Knocks. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3764–3774. https://doi.org/10.1145/3025453.3025991

[6] Adrian A. de Freitas, Michael Nebeling, Xiang 'Anthony' Chen, Junrui Yang, Akshaye Shreenithi Kirupa Karthikeyan Ranithangam, and Anind K. Dey. 2016. Snap-To-It: A User-Inspired Platform for Opportunistic Device Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 5909–5920. https://doi.org/10.1145/2858036.2858177

[7] Android Developers. 2018. Guides for Android audio latency. Retrieved September 20, 2018 from https://developer.android.com/ndk/guides/audio/audio-latency.html.

[8] Jeff Dunn. 2017. It looks like Apple has some work to do if it wants Siri to be as smart as Google Assistant. Retrieved September 20, 2018 from https://goo.gl/4spfhy.

[9] Mingming Fan, Alexander Travis Adams, and Khai N. Truong. 2014. Public Restroom Detection on Mobile Phone via Active Probing. In *Proceedings of the 2014 ACM International Symposium on Wearable Computers (ISWC '14)*. ACM, New York, NY, USA, 27–34. https://doi.org/10.1145/2634317.2634320

[10] Taesik Gong, Jun Hyuk Chang, Joon-Gyum Kim, Soowon Kang, Donghwi Kim, and Sung-Ju Lee. 2017. Enjoy the Silence: Noise Control with Smartphones. In *Computer Communication and Networks (ICCCN), 2017 26th International Conference on*. IEEE, 1–9.

[11] Taesik Gong, Hyunsung Cho, Bowon Lee, and Sung-Ju Lee. 2018. Identifying Everyday Objects with Smartphone Knock. In *Proceedings of the 2018 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM.

[12] Daniel Groeger and Jürgen Steimle. 2018. ObjectSkin: Augmenting Everyday Objects with Hydroprinted Touch Sensors and Displays. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 134 (Jan. 2018), 23 pages. https://doi.org/10.1145/3161165

[13] Anhong Guo, Xiang 'Anthony' Chen, Haoran Qi, Samuel White, Suman Ghosh, Chieko Asakawa, and Jeffrey P. Bigham. 2016. VizLens: A Robust and Interactive Screen Reader for Interfaces in the Real World. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 651–664. https://doi.org/10.1145/2984511.2984518

[14] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter* 11, 1 (2009), 10–18.

[15] Chris Harrison, Julia Schwarz, and Scott E. Hudson. 2011. TapSense: Enhancing Finger Interaction on Touch Surfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 627–636. https://doi.org/10.1145/2047196.2047279

[16] Chris Harrison, Robert Xiao, and Scott Hudson. 2012. Acoustic Barcodes: Passive, Durable and Inexpensive Notched Identification Tags. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 563–568. https://doi.org/10.1145/2380116.2380187

[17] Hernisa Kacorri, Kris M. Kitani, Jeffrey P. Bigham, and Chieko Asakawa. 2017. People with Visual Impairment Training Personal Object Recognizers: Feasibility and Challenges. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 5839–5849. https://doi.org/10.1145/3025453.3025899

[18] Jacob Kastrenakes. 2017. Burger King's new ad forces Google Home to advertise the Whopper. https://www.theverge.com/2017/4/12/15259400/burger-king-google-home-ad-wikipedia

[19] Beth M. Lange, Mark A. Jones, and James L. Meyers. 1998. Insight Lab: An Immersive Team Environment Linking Paper, Displays, and Data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '98)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 550–557. https://doi.org/10.1145/274644.274718

[20] Gierad Laput, Walter S. Lasecki, Jason Wiese, Robert Xiao, Jeffrey P. Bigham, and Chris Harrison. 2015. Zensors: Adaptive, Rapidly Deployable, Human-Intelligent Sensor Feeds. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1935–1944. https://doi.org/10.1145/2702123.2702416

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

[22] Gierad Laput, Chouchang Yang, Robert Xiao, Alanson Sample, and Chris Harrison. 2015. EM-Sense: Touch Recognition of Uninstrumented, Electrical and Electromechanical Objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology (UIST '15)*. ACM, New York, NY, USA, 157–166. https://doi.org/10.1145/2807442.2807481

[23] Gierad Laput, Yang Zhang, and Chris Harrison. 2017. Synthetic Sensors: Towards General-Purpose Sensing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3986–3999. https://doi.org/10.1145/3025453.3025773

[24] Hanchuan Li, Can Ye, and Alanson P. Sample. 2015. IDSense: A Human Object Interaction Detection System Based on Passive UHF RFID. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2555–2564. https://doi.org/10.1145/2702123.2702178

[25] Pedro Lopes, Ricardo Jota, and Joaquim A. Jorge. 2011. Augmenting Touch Interaction Through Acoustic Sensing. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '11)*. ACM, New York, NY, USA, 53–56. https://doi.org/10.1145/2076354.2076364

[26] Shan Luo, Leqi Zhu, Kaspar Althoefer, and Hongbin Liu. 2017. Knock-Knock: Acoustic object recognition by using stacked denoising autoencoders. *Neurocomput.* 267, C (Dec. 2017), 18–24. https://doi.org/10.1016/j.neucom.2017.03.014

[27] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[28] Takuya Maekawa, Yasue Kishino, Yasushi Sakurai, and Takayuki Suyama. 2011. Recognizing the Use of Portable Electrical Devices with Hand-Worn Magnetic Sensors. In *Pervasive Computing*, Kent Lyons, Jeffrey Hightower, and Elaine M. Huang (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 276–293.

[29] T. Maekawa, Y. Kishino, Y. Yanagisawa, and Y. Sakurai. 2012. WristSense: Wrist-worn sensor device with camera for daily activity recognition. In *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*. 510–512. https://doi.org/10.1109/PerComW.2012.6197551

[30] Kenneth Olmstead. 2017. Nearly half of Americans use digital voice assistants, mostly on their smartphones. Retrieved September 20, 2018 from https://goo.gl/gRyF4R.

[31] Jun Rekimoto and Yuji Ayatsuka. 2000. CyberCode: Designing Augmented Reality Environments with Visual Tags. In *Proceedings of DARE 2000 on Designing Augmented Reality Environments (DARE '00)*. ACM, New York, NY, USA, 1–10. https://doi.org/10.1145/354666.354667

[32] X. Ren and M. Philipose. 2009. Egocentric recognition of handled objects: Benchmark and analysis. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 1–8. https://doi.org/10.1109/CVPRW.2009.5204360

[33] Nirupam Roy, Sheng Shen, Haitham Hassanieh, and Romit Roy Choudhury. 2018. Inaudible Voice Commands: The Long-Range Attack and Defense. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*. USENIX Association, 547–560.

[34] Lei Shi, Maryam Ashoori, Yunfeng Zhang, and Shiri Azenkot. 2018. Knock Knock, What's There: Converting Passive Objects into Customizable Smart Controllers. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '18)*. ACM, New York, NY, USA, Article 31, 13 pages. https://doi.org/10.1145/3229434.3229453

[35] M. Strese, C. Schuwerk, A. Iepure, and E. Steinbach. 2017. Multimodal Feature-Based Surface Material Classification. *IEEE Transactions on Haptics* 10, 2 (April 2017), 226–239. https://doi.org/10.1109/TOH.2016.2625787

[36] Android Studio. 2019. Profile battery usage with Batterystats and Battery Historian. Retrieved May 10, 2019 from https://developer.android.com/studio/profile/battery-historian.

[37] Naomi van der Velde. 2018. A Complete Speech Recognition Technology Overview. Retrieved September 20, 2018 from https://www.globalme.net/blog/the-present-future-of-speech-recognition.

[38] Edward J. Wang, Tien-Jui Lee, Alex Mariakakis, Mayank Goel, Sidhant Gupta, and Shwetak N. Patel. 2015. MagnifiSense: Inferring Device Interaction Using Wrist-worn Passive Magneto-inductive Sensors. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)*. ACM, New York, NY, USA, 15–26. https://doi.org/10.1145/2750858.2804271

[39] Roy Want, Kenneth P. Fishkin, Anuj Gujar, and Beverly L. Harrison. 1999. Bridging Physical and Virtual Worlds with Electronic Tags. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, USA, 370–377. https://doi.org/10.1145/302979.303111

[40] Robert Xiao, Gierad Laput, Yang Zhang, and Chris Harrison. 2017. Deus EM Machina: On-Touch Contextual Functionality for Smart IoT Appliances. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 4000–4008. https://doi.org/10.1145/3025453.3025828

[41] C. Yang and A. P. Sample. 2016. EM-ID: Tag-less identification of electrical devices via electromagnetic emissions. In *2016 IEEE International Conference on RFID (RFID)*. 1–8. https://doi.org/10.1109/RFID.2016.7488014

[42] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, Xiaofeng Wang, and Carl A Gunter. 2018. CommanderSong: A Systematic Approach for Practical Adversarial Voice Recognition. *arXiv preprint arXiv:1801.08535* (2018).

[43] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2017. DolphinAttack: Inaudible Voice Commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*. ACM, New York, NY, USA, 103–117. https://doi.org/10.1145/3133956.3134052

[44] Fang Zheng, Guoliang Zhang, and Zhanjiang Song. 2001. Comparison of different implementations of MFCC. *Journal of Computer Science and Technology* 16, 6 (01 Nov 2001), 582–589. https://doi.org/10.1007/BF02943243