Video: Real-Time Object Identification with a Smartphone Knock

Taesik Gong
taesik.gong@kaist.ac.kr
KAIST

Hyunsung Cho
hyunsungcho@kaist.ac.kr
KAIST

Bowon Lee
bowon.lee@inha.ac.kr
Inha University

Sung-Ju Lee
profsj@kaist.ac.kr
KAIST

ABSTRACT

We propose Knocker, a real-time object identification technique with smartphones. Knocker leverages unique impulse signals that are generated by knocking on an object with a smartphone. Knocker does not require any special augmentation for both smartphones and objects.

CCS CONCEPTS

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

ACM Reference Format:
Taesik Gong, Hyunsung Cho, Bowon Lee, and Sung-Ju Lee. 2019. Video: Real-Time Object Identification with a Smartphone Knock. In *The 17th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '19), June 17–21, 2019, Seoul, Republic of Korea.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3307334.3328593

1 INTRODUCTION

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. 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 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 rooting from innate complexity of natural languages , and privacy and security concerns. Identifiable artificial markers are most commonly used to identify objects for interaction, such as barcodes, QR codes, and RFID tags [3]. 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 [1] to electromagnetic (EM) based sensing [2] 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.

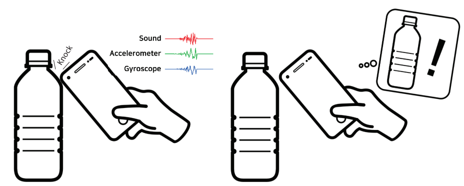

Figure 1: An example knock on a bottle.

We introduce Knocker[1] 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.

2 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.

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

MobiSys '19, June 17–21, 2019, Seoul, Republic of Korea

ACM ISBN 978-1-4503-6661-8/19/06.
https://doi.org/10.1145/3307334.3328593

[1]Video link: https://www.youtube.com/watch?v=Nacs204Z2JA

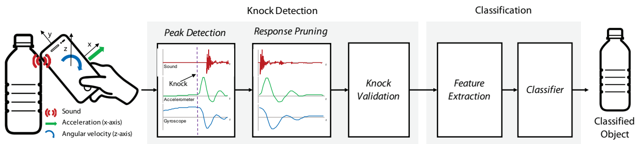

Figure 2: Knocker system overview.

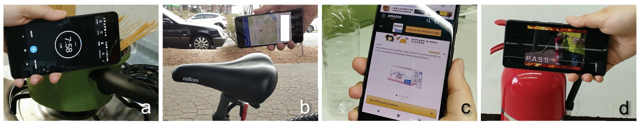

Figure 3: Object-specific applications.

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. 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 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.

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 3(a)). For a bicycle rider, Knocker can record the latest parking location when the user knocks on the saddle (Figure 3(b)). Knocking on consumer goods can launch an

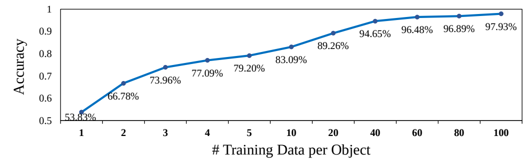

Figure 4: Accuracy as a function of the amount of training data per object. Per-user classifiers are used.

appropriate e-commerce app or website and retrieves the result of searching the object (Figure 3(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 3(d)).

4 EVALUATION

As Knocker targets everyday objects that a person regularly interacts with, we evaluate Knocker's performance using per-user classifiers. For each user, we use 100 knocks for training and 10 for testing per each object (total 2,300 for training and 230 for testing). 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 4). As expected, accuracy improves with more training data. Interestingly, only five knocks per object achieves nearly 80% accuracy with 23 objects. With 60 knocks per object, Knocker exceeds 96% 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 (STDEV = 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.

ACKNOWLEDGEMENT

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No.2016R1A2B4014068), Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2017M3C4A7083534), and the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2018S1A5A2A03037308).

REFERENCES

[1] 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

[2] 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

[3] 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