Ubicoustics: Plug-and-Play Acoustic Activity Recognition

Gierad Laput Karan Ahuja Mayank Goel Chris Harrison
Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213
{gierad.laput, kahuja, mayank, chris.harrison}@cs.cmu.edu

Figure 1. Ubicoustics is an activity sensing system that takes advantage of the ubiquity of microphones in modern consumer electronics. Our system starts with state-of-the-art sound labeling models trained on millions of online videos, which are tuned to classes of interest using data from professional sound effect libraries. This enables real-time activity recognition without any end user or *in situ* training, across diverse hardware platforms, including smart speakers (A), smartwatches (B), tablets (C), phones (D), IoT sensors (E) and laptops (F).

ABSTRACT
Despite sound being a rich source of information, computing devices with microphones do not leverage audio to glean useful insights about their physical and social context. For example, a smart speaker sitting on a kitchen countertop cannot figure out if it is in a kitchen, let alone know what a user is doing in a kitchen – a missed opportunity. In this work, we describe a novel, real-time, sound-based activity recognition system. We start by taking an existing, state-of-the-art sound labeling model, which we then tune to classes of interest by drawing data from professional sound effect libraries traditionally used in the entertainment industry. These well-labeled and high-quality sounds are the perfect atomic unit for data augmentation, including amplitude, reverb, and mixing, allowing us to exponentially grow our tuning data in realistic ways. We quantify the performance of our approach across a range of environments and device categories and show that microphone-equipped computing devices already have the requisite capability to unlock real-time activity recognition comparable to human accuracy.

Author Keywords
Ubiquitous sensing; IoT; Smart Environments;

CCS Concepts
Human-centered computing~ Ubiquitous and mobile computing systems and tools.

INTRODUCTION
Microphones are the most common sensor found in consumer electronics today, from smart speakers and phones to tablets and televisions. Despite sound being an incredibly rich information source, offering powerful insights about physical and social context, modern computing devices do not utilize their microphones to understand what is going on around them. For example, a smart speaker sitting on a kitchen countertop cannot figure out if it is in a kitchen, let alone know what a user is doing in a kitchen (Figure 1A). Likewise, a smartwatch worn on the wrist is oblivious to its user cooking or cleaning (Figure 1B). This inability for "smart" devices to recognize what is happening around them in the physical world is a major impediment to them truly augmenting human activities.

Real-time, sound-based classification of activities and context is not new. There have been many previous application-specific efforts that focus on a constrained set of recognized classes [18, 35, 42, 41]. For example, Ward *et al.* [41] developed a microphone-equipped necklace in conjunction with accelerometers mounted on arms that could distinguish between nine shop tools. In these types of constrained uses, the training data for machine learning is generally domain-specific and captured by the researchers themselves.

We sought to build a more general-purpose and flexible sound recognition pipeline – one that could be deployed to an existing device as a software update and work immediately, requiring no end-user or *in situ* data collection (*i.e.*, no training or calibration). Such a system should be "plug-and-play" – *e.g.*, plug in your Alexa, and it can immediately discern all of your kitchen appliances by sound. This is a challenging task, and very few sound-based recognition systems achieve usable end-user accuracies, despite offering pre-trained models that are meant to be integrated into applications (*e.g.*, Youtube-8M [2], SoundNet [4]).

We propose a novel approach that brings the vision of plug-and-play activity recognition closer to reality. Our process starts by taking an existing, state-of-the-art sound labeling model and tuning it with high-quality data from professional sound effect libraries for specific contexts (*e.g.*, a kitchen and its appliances). We found professional sound effect libraries to be a particularly rich source of high-quality, well-segmented, and accurately-labeled data for everyday events.

UIST '18, October 14-17, 2018, Berlin, Germany
© 2018 ACM. ISBN 978-1-4503-5948-1/18/10…$15.00
https://doi.org/10.1145/3242587.3242609

These large databases are employed in the entertainment industry for post-production sound design (and to a lesser extent in live broadcast and digital games).

Sound effects can also be easily transformed into hundreds of realistic variations (synthetically growing our dataset, as opposed to finding or recording more data) by adjusting key audio properties such as amplitude and persistence, as well mixing sounds with various background tracks. We show that models tuned on sound effects can achieve superior accuracy to those trained on internet-mined data alone. We also evaluate the robustness of our approach across different physical contexts and device categories. Results show that our system can achieve human-level performance, both in terms of recognition accuracy and false positive rejection.

Overall, this paper makes the following contributions in the area of activity recognition and ubiquitous sensing:

1. We developed a real-time, activity recognition system that demonstrates accuracies and class diversity approaching end user feasibility, requiring no in-situ data collection, and using nothing but commodity microphones for input.
2. We ran a comprehensive suite of experiments that quantify the performance of our system across 4 data augmentations, 7 device categories, 7 location contexts, and 30 recognition classes, providing insights into the feasibility of sound-based activity recognition that generalize beyond our implementation.
3. In addition to conventional testing with existing sound datasets, we move beyond prior work by capturing a new, real-world dataset with improved ecological validity. We also benchmark our results against human accuracy (600 participants) to better contextualize performance.
4. Finally, we share our data, processing pipeline and trained models to facilitate replication and new uses.

WHY SOUND EFFECTS?

Sound effect libraries have several fundamental properties that make them ideal for training machine learning models, which we now describe.

Properties of Sound Effects

First, sound effects are ***atomic*** – a clip labeled as "door knock" or "cat meow" is tightly segmented and contains only that one sound. Sound effects are also ***pure*** – clips are generally recorded in professional studios and are devoid of artifacts like background noise and echoes. Such purity is mandatory, as the sounds are meant to be transformed and composited into richer soundscapes. Third, sound effect libraries are ***diverse***; post-production sound editors search for the perfect sound based on the materials in the scene and mood of the shot. For this reason, libraries often contain hundreds of variations of the same sound effect.

Properties of Sound

Sound itself has three important properties. Foremost, sound data is ***scalable***, both in amplitude and duration. Second, sounds are ***transformable***, able to be projected into synthetic environments by altering their equalization ("EQ"), reverb and damping (*i.e.*, persistence). In this manner, an effect can be made to sound like it is in a furnished living room or small bathroom. Finally, audio is innately ***additive*** (though not subtractive), allowing two or more effects to be trivially blended. We can take a sound effect and trivially combine it with an ambient track of a bustling market, tranquil forest, or ambient hum of HVAC.

Taken together, this means we can take a *single* sound effect, and transmute it into *hundreds* of realistic variations. When applied to entire sound effect libraries, we can achieve a scale of data ideal for training deep learning models, while retaining all of the benefits of a highly-curated corpus. We show in our subsequent evaluation that such models outperform those trained on comparably-sized, internet-mined data.

Categories of Sound Effects

There are three main categories of sound effects: *hard*, *natural*, and *background* sounds [30]. Hard sounds are closely linked to on-screen action (*e.g.*, cough, door closing). Natural sounds are subtle effects that add realism to a scene and action (*e.g.*, leaves rustling, fabric chafing). Finally, background sounds (*e.g.*, HVAC hum, engine noise) are used to build immersive soundscapes, smooth breaks in dialogue, and anchor visual transitions. In Ubicoustics, we rely on *hard* sounds for our training data and use *background* sounds for our mix augmentations, described later.

We note that many professional sound effects are produced through Foley – the recreation or simulation of a sound. In interviews with sound production and Foley artists (with on-screen credits) conducted as part of background research, we found that Foley is generally used when scenes require precise audio-visual timing (*e.g.*, person walking on snow) or for actions where natural examples are rare (*e.g.*, blood spatter). We confirmed that the commonplace environmental sounds that we focus on are rarely Foleyed, as these are easier and cheaper to record than simulate (*e.g.*, coughing, toilet flush, blender, vacuum).

RELATED WORK

There are many approaches for sensing human activity, from special-purpose sensors such as geophones [17], water pressure sensors [12], powerline sensing [15] and RF tags [21], to generic approaches such as computer vision [19]. In this section, we focus on sound-driven methods that most closely relate to our efforts.

Sound Event Classification

There are a number of machine learning models that leverage publicly-available audio datasets to classify sound events. For instance, Salamon *et al.* [33] employed scattering transform for urban sound classification (classes include jackhammer and car horn), while Foggia *et al.* [11] employ a bag of words-based approach with a multi-class SVM for detection for surveillance uses (*e.g.*, screaming, glass breaking, gunshot). Sound has also been used for scene and context recognition, for example, Eronen *et al.* [10] used

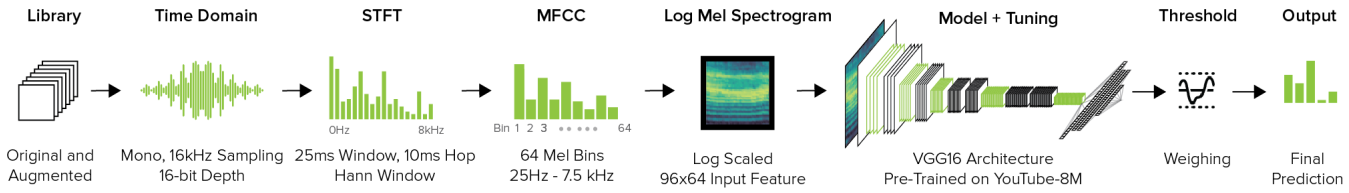

Figure 2. Ubicoustics process overview. We use a corpus of augmented sound effects to tune a state-of-the-art sound model (*e.g.*, YouTube-8M, trained on 8 million videos) for a particular end user application. We perform a context-based confidence thresholding to improve robustness, especially for "unknown" sounds.

clustering and hidden Markov models for audio context recognition (including outdoors, vehicles and homes).

More recently, deep learning has been applied to sound event classification. For example, Lane *et al.* leverage several fully connected layers for audio sensing in DeepEar [18], which focused on classification of high-level categories. The closest to Ubicoustics is a four-class "voicing, music, water and traffic" set, which is quite different from the fine-grained activity classes we aim to support (toilet flushing, chopping, knocking, coughing, microwaving, typing, etc.). SoundNet [4] used video data and computer vision to identify objects in a scene, and then used the resulting labels to learn sounds.

Convolutional neural networks (CNNs) have also been employed for sound event classification. McLoughlin *et al.* [24], Ephrat *et al.* [8] and Phan *et al.* [28] present CNN-based deep learning architectures, while Parascandolo *et al.* [27] use Convolutional Recurrent Neural Networks (CRNNs). Hershey *et al.* [16] compare various CNN architectures for acoustic event detection and benchmark on the AudioSet dataset [13]. The Never-Ending Learner of Sounds (NELS) [8] crawls the web, continuously training a CNN using semi-structured online data, creating an index of sounds. We leveraged this prior work heavily in the design of our system.

Sound data augmentation has also been previously explored to improve the robustness of acoustic classification models. For example, McFee *et al.* [23] used pitch shift, time stretch, dynamic range compression, and background noise addition for music. Salamon *et al.* [32] used the same augmentation techniques in conjunction with a CNN, evaluating on the UrbanSound8K dataset. We drew inspiration from these prior efforts when developing our own set of augmentations.

Real-Time, Sound-Driven Activity Recognition

Beyond "offline" event labeling, sound has long been used for inferring *real-time activities*. Likely the most pervasive system is ShotSpotter [35], which provides gunshot detection and localization for law enforcement. For a more general set of activities, Stork *et al.* [37] used Mel-frequency cepstral coefficients (MFCC) with non-Markovian ensemble voting to discriminate among 22 human activities from bathroom and kitchen contexts (including blender mixing, pouring water, sorting dishes). Lu *et al.* [22] demonstrated speech, music and ambient sound detection using a phone microphone. Synthetic Sensors [20] used a custom board equipped with acoustic sensing capabilities (among many sensor channels) to distinguish 38 environmental events. There have also been portable systems using body-worn microphones and accelerometers, including the Mobile Sensing Platform [40], BodyScope [42] and work by Ward *et al.* [41]. This prior work requires training within a user's environment and focuses on specific domains and devices.

UBICOUSTICS

We now describe our process for enabling ubiquitous acoustic activity sensing, which is illustrated in Figure 2.

Contexts and Classes

The first step is defining a context of use (*e.g.*, construction site, hospital, dentist office, café), which limits the set of classes to be recognized. These classes can then be mined from sound effect libraries, often just by using the name of the class itself as the search term, though we found more clips can be identified with some keyword variation (*e.g.*, not just "faucet", but also "water running", "water flowing", "tap dripping", etc.). The result is a large corpus of sound effects covering the classes of interest.

Sound Pre-Processing

Once the corpus is assembled, we standardize all sounds into a single format, as libraries come in different file formats (*e.g.*, WAVs, AIFFs), bit depths (8 to 32 bits), sample rates (16-48 kHz), and number of channels (mono to 5.1 surround sound). We selected 16 kHz mono (16-bit) as our standard format, as we found this to be the lowest common denominator. Once converted, we removed silences greater than one second anywhere in the clip. At this point, we have what we call an "*original*" sound set (*i.e.*, no augmentations).

Amplification Augmentation

To begin to add variation to a sound dataset, we first apply an amplitude augmentation. We produce two variations for each input sound (Figure 3), one quieter (25% of original volume) and one louder (raising peak amplitude to -0.1dB).

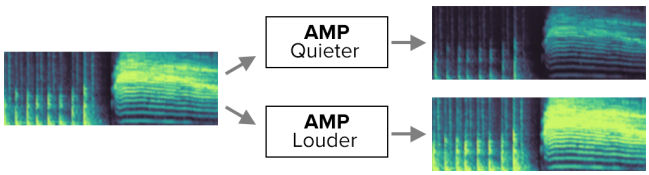

Figure 3. Amplifications (quiet, loud) applied to a sound effect.

Persistence Augmentation

Our next augmentation modifies the persistence of a sound effect (Figure 4), which includes reverberations and non-linear damping (see *e.g.*, [7, 31] for more background). By

modifying these parameters, we can simulate sounds in a variety of physical spaces (*e.g.*, kitchen, hallway, bathroom).

We used two methods to generate realistic persistence transformations. First, we selected six professional "reverb" effects provided by Adobe Audition (Figure 6, grey dots). Second, we created four custom convolutional reverbs by capturing impulse functions [3] in exemplary rooms: a bathroom, large atrium, workshop and small office (Figure 6, green dots). To create these custom effects, we placed a speaker/microphone setup in a target room. We then emitted a sinusoidal sweep and recorded the frequency response, which we de-convolved into an impulse function. We plot our ten persistence transformations in Figure 6 by the size of room and absorption level. In total, this augmentation process yields ten new sounds for every input sound.

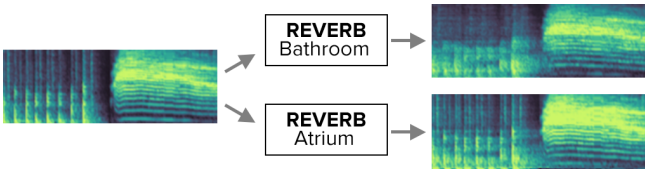

Figure 4. Custom bathroom and atrium impulse functions, and their resulting persistence of sound augmentations.

Mixing Augmentation

Our next augmentation blends sound effects with background sounds we sourced. This mixing process introduces foreign elements to original sounds, adding variability (Figure 5). Each input sound is mixed with a randomly selected background segment, which includes indoor (*e.g.*, HVAC), outdoor (*e.g.*, birds chirping), urban (*e.g.*, vehicle traffic), and social (*e.g.*, cafe) background noise. In this way, we create six new sounds for every input sound.

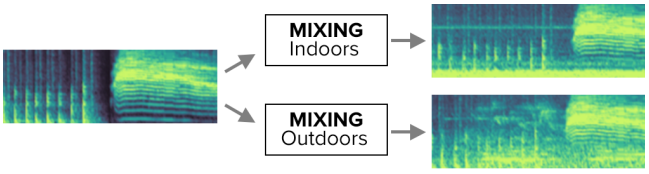

Figure 5. Mixing augmentation applied to a sound effect: indoor (top), and outdoor (bottom) ambient sounds.

Combining Augmentations

Finally, we can stack and combine augmentations, creating even greater variety. For example, we can take a "brushing teeth" sound effect and make it louder, apply a bathroom-like reverberation, and add background noise from an exhaust fan. Note the order of operations is important. For instance, background tracks generally already include reverb, as they are recorded on location, and thus re-projecting them into a second environment leads to less realistic output.

Featurization

Once the sound dataset is assembled, we compute its features. There are many existing featurization stacks for audio data; in our implementation, we chose the method described in Hershey *et al.* [16]. First, we segment files into 960 ms audio segments and compute short-time Fourier Transforms for each segment (using a 25 ms window and step size of 10 ms), which yields a 96-length spectrogram. We then convert our linear spectrogram into a 64-bin log-scaled Mel spectrogram and generate a 96×64 input frame for every 960 ms of audio (see Figure 7), which is fed into our classification model.

Model Architecture

We build upon the YouTube-8M VGG-16 [16] model, which is a variant of the VGG16 architecture trained on 8 million YouTube videos. The architecture contains four convolutional layers (3×3 kernel, step size = 2, depth = 64, 128, 256, and 512, ReLU activation [25]), with intermediary max pool layers [14], and a 128-wide fully connected embedding layer. We modified this pre-trained model by removing the last fully connected layer and replacing it with our own fully connected layer, using a sigmoid activation function. Finally, we tune the entire architecture with our sound effect datasets.

Devices

In a commercial implementation, we envision models running locally on devices, as opposed to streaming data to cloud infrastructure as is common today. Local classification has obvious latency and privacy benefits. As a proof of concept, we deployed our model to three exemplary devices spanning a range of computational abilities: Apple MacBook Pro 2017, iPhone 7 smartphone, and Raspberry Pi Zero W with a ReSpeaker dual mic shield [34]. The models run at 15.2, 8.3 and 0.7 frames per second on these devices respectively. This performance is already sufficiently granular for most of the activities we studied (which last on the order of seconds), and suggests that with additional engineering and optimization, interactivity on embedded devices is possible.

Open Model and Data

We make our training datasets, architecture, trained models, visualization tools and data collection pipeline available for researchers and practitioners to build upon. It consists of processed Mel spectrograms of our original and augmented sound effects. We also include the raw WAV files and processed Mel spectrograms for sounds captured in our "in-the-wild" data collection across seven devices (see Evaluation section). http://www.github.com/FIGLAB/ubicoustics

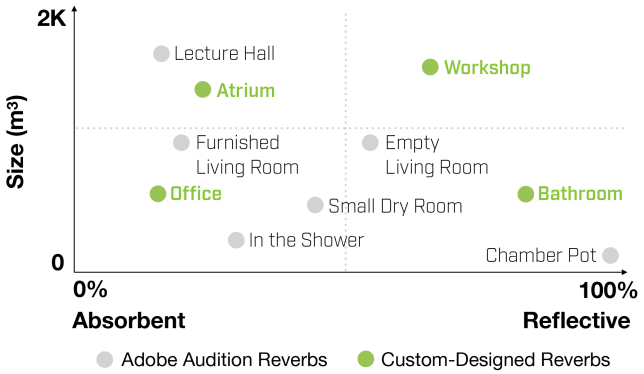

Figure 6. Persistence of sound transformations plotted by room size and sound absorption.

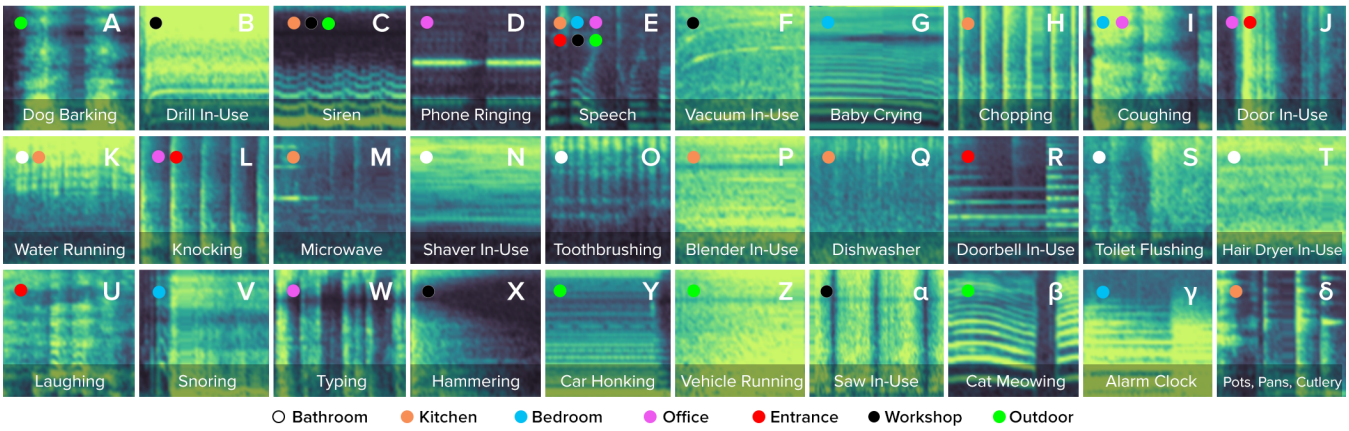

Figure 7. Example log Mel spectrograms (based on [16]) of 960ms audio for our 30 test event classes. These 96×64 input vectors are used to tune a VGG-16 model pre-trained on YouTube-8M.

EVALUATION

We sought to answer several key questions: What is the performance of a classifier tuned with sound effects? Does sound augmentation improve performance? How well does the model perform when tested on live, real-world data? Does the model work across different devices? And how does our technique compare to human accuracy?

Contexts

For our evaluation, we selected seven location contexts in which everyday activities occur: 1) bathroom, 2) bedroom, 3) house entrance, 4) kitchen, 5) office, 6) outdoor and 7) workshop. These contexts offer realistic scenarios with constrained event classes. For instance, it is highly unlikely for a "blender" event to occur in a bathroom, or for "chopping" to happen in a bedroom. This permits us to tune models for particular contexts with more tractable class sets.

Classes

For each context, we selected commonplace events using the following selection criteria: a) does the event happen frequently in that context? b) does it produce enough acoustic energy to be heard by a microphone? and c) can knowledge of the event enable useful applications? In total, we selected 30 events across our seven test contexts (Figure 7).

Sound Sources

There are dozens of large sound effect libraries to draw upon for tuning data. As a representative cross-section, we selected five libraries that were available online or licensed by our institution, listed in Table 1.

Name	Total Sounds	Sounds Used	Hours Used
BBC Sound Library [5]	29K	740	1.9
Network Sound Library [26]	10K	492	1.3
Soundsnap [36]	250K	4072	10.4
FreeSound [43]	372K	8929	22.2
AudioSet [13]	2000K	7899	18.2

Table 1. The sound sources we used for our datasets.

Tune and Test Sets

From the sources listed in Table 1, we extracted sound data covering our 30 classes. We split this corpus into train (*i.e.*, tune) and test sets: *SFX-Orig* and *SFX-Test*. Our models are also tuned on different augmentations of *SFX-Orig*: amplification (*SFX-Amp*), persistence of sound (*SFX-Persist*), and mixing (*SFX-Mix*). We also created a corpus containing all augmentations (*SFX-All*), comprising almost 500 hours of sounds clips for our 30 classes. These various datasets are summarized in Table 2.

Set Name	Contains	Data Hours
SFX-Orig	Processed, but otherwise unaugmented sounds	54.6
SFX-Amp	SFX-Orig + Amplitude Augmentations	152.0
SFX-Persist	SFX-Orig + Persistence Augmen. (15% draw)	136.2
SFX-Mix	SFX-Orig + Mix Augmentations (75% draw)	300.6
SFX-All	SFX-Orig + SFX-Amp + SFX-Persist + SFX-Mix	479.5
SFX-Test	Unaugmented sounds; holdout test set.	8.8
In-the-Wild Test	Sounds recorded on seven exemplary devices (see Table 3); holdout test set.	12.3

Table 2. Summary of datasets we created for our evaluations.

Test Devices

In order to test the robustness of our models across different microphones, placements, and platforms, we developed software to capture and stream audio from a diverse set of hardware platforms (Table 3). These devices connected over various means to an independent laptop capable of recording synchronized streams and performing live classification.

Device Type	Implementation	Comm.
Smartphone *(two placements)*	iPhone 5C, Swift iOS app	Wi-Fi
Smartwatch	LG W100, Android app	Bluetooth
Smart speaker	Jabra Speak 410	USB
IoT Sensor	Custom hardware	Bluetooth
Laptop	MacBook Pro 2013, Python app	Wi-Fi
Tablet	iPad 3, Swift iOS app	Wi-Fi

Table 3. Devices we used to capture data for our *In-The-Wild Test* set. Two identical smartphones were used to record on-table and in-pocket data.

Collecting In-the-Wild Sounds

In addition to evaluating our models on *SFX-Test* (8.8 hours of mined sound effects), we also wished to test on more ecologically valid "in-the-wild" dataset, captured not in a studio, but with microphones found in real-world devices recording in real-world environments. In response, we recruited 12 participants (mean age 29.3) who performed or triggered events across 50 rooms, spanning dozens of homes and buildings. The experimenter used a laptop to synchronously capture audio data from our seven test devices (Table 3). The interface allowed the experimenter to demarcate the start and end of events, as well as enter a ground truth label.

When collecting data in a room (*e.g.*, kitchen 5), devices were placed in a realistic fashion. For example, the laptop, tablet and smart speaker were placed on a logical flat surface, while participants wore the smartwatch, and the IoT sensor was plugged-in to a nearby power outlet. For the smartphone category, we captured data for two placements (using two identical phones): a) phone in a participant's pocket, and b) phone on a surface. Devices were never more than 3 meters from an event source, and we avoided making changes to the physical layout of the space (no special tables, no appliances moved, etc.).

In each location, we collected three rounds of data per event, in a random order. Sometimes this was activating an appliance (*e.g.,* running a microwave) while other times it was a physical task (*e.g.,* chopping vegetables). In all cases, the materials and equipment were participants' own. For events such as "coughing" and "laughing," we asked users to perform the action as naturally as possible. We excluded events that were challenging to induce (*e.g.*, baby crying, hazard alarm). All data was collected between 10am and 8pm; other occupants were free to go about their daily routines, which injected some natural noise. In total, we collected 12.3 hours of labeled/segmented data, which we call *In-The-Wild Test*.

RESULTS AND DISCUSSION

We now describe the results from a series of integrated experiments. A summary of main study results can be found in Figures 8, 9, 13 and 14. First, we evaluate the "plug-and-play" accuracy of Ubicoustics, including rejection of unknown sounds. We then compare performance to human annotators, which serve as a gold standard. Finally, we investigate the effects of various augmentations, device categories, and location contexts.

Accuracy

For all accuracy metrics, we use clip-level prediction. More concretely, we record a model's output across an entire sound clip and return the top predicted result, based on cumulative confidence. As noted above, we use a dedicated model for each context (tuned only for classes that belong to that particular context). At the end of every tuning epoch, we checkpoint the model against both the *SFX-Test* and *In-the-Wild Test* datasets respectively and report the accuracy of the best performing epoch (a common, but artificial method we improve upon in the next section).

Overall, per-context models tuned on *SFX-All* and tested on *SFX-Test* achieved an average accuracy of 93.9% (SD=3.7%; Figure 8, *SFX-All*, green bar). When tested on *In-the-Wild Test,* average accuracy dropped to 89.6% (SD=6.3%; Figure 8, *SFX-All*, blue bar). When we tune our model using only AudioSet data for our classes, the system achieves an average accuracy of 70.6% and 69.5% on *SFX-Test* and *In-the-Wild Test,* respectively (Figure 8, AudioSet). This result underscores the significant boost in accuracy when tuning on sound effect libraries. If we disregard context, and tune/test on all 30 classes, our *SFX-All* tuned model achieves an accuracy of 82.1% and 68.4% on *SFX-Test* and *In-The-Wild Test* respectively.

Better Estimating Real-World Accuracy

Although the aforementioned 89.6% accuracy follows a standard evaluation procedure, it does not offer a fair depiction of "plug-and-play" accuracy, as one would experience in a real-world deployment. Foremost, when deploying to, *e.g.*, a user's home, there is no test data on which to checkpoint model training. Second, real-world deployments are subjected to "unknown" sounds, never before heard by the classifier. Evaluating models using only classes they know offers no insights into how interactive systems will handle false positive events. Thus, we ran two additional experiments to more conservatively estimate our system's performance.

First, we tuned per-context models using *SFX-All*, checkpointed on *SFX-Test,* and evaluated on *In-the-Wild Test*. This procedure inherently removes checkpointing bias. Using this more stringent procedure, average accuracy across per-context classifiers was 84.8% (SD=6.6%).

As a second, even harder test, we devised an experiment that included 20% "unknown" classes (*i.e.*, sounds we drew from other contexts) in the test set that the model should ignore. This required some alterations to our pipeline. Instead of many per-context models, we tuned a single model using data from all 30 classes in *SFX-All*. A sound is classified as "unknown" (and ignored) if no in-context class exceeds a confidence threshold. Using this evaluation procedure on *In-the-Wild Test* (checkpointed on *SFX-Test*), we found an across-context accuracy of 80.4% (SD=6.4%; Figure 9, *SFX-All*), which we believe is a much closer estimation of plug-and-play performance. See Figures 11 and 12 for the confusion matrices for these two experiments.

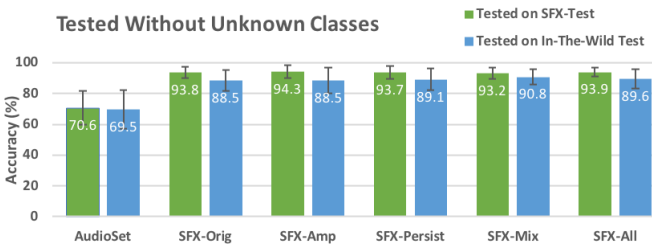

Figure 8. Recognition accuracies when evaluated without any unknown classes and checkpointed on best model.

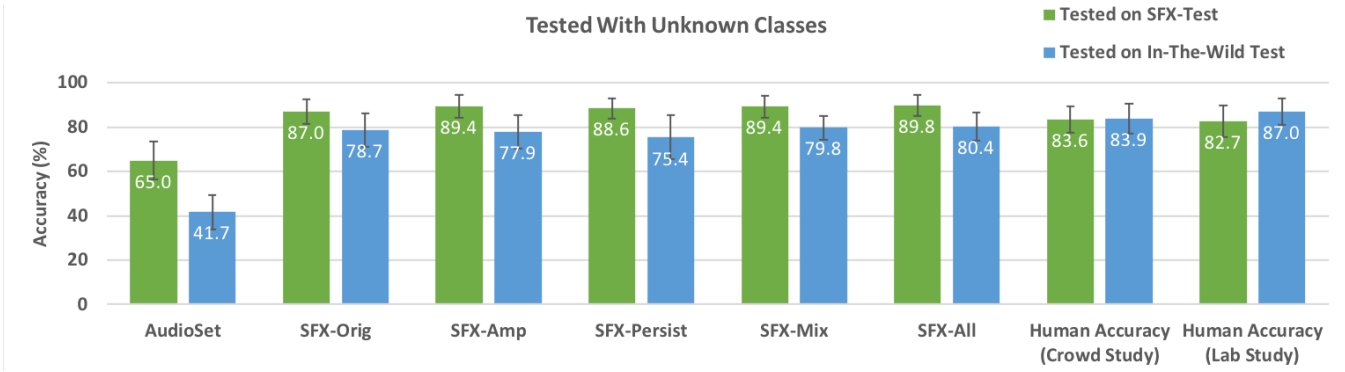

Figure 9. Recognition accuracies when evaluated with unknown classes comprising 20% of the test data. *In-the-Wild Test* is checkpointed on *SFX-Test* to remove checkpoint bias (*SFX-Test* is checkpointed on *SFX-Test*).

Comparison to Human Performance

While 100% accuracy is the ultimate goal of any interactive system, some problem domains are particularly ambiguous or challenging and can benefit from additional baselines to contextualize performance. In the case of sound classification, humans offer an excellent gold standard, as they can draw upon a lifetime of real-world experiences and leverage contextual knowledge in sophisticated ways (*e.g.,* a small motorized appliance in a kitchen is more likely to be a blender than a miter saw). As such, we conducted two studies to establish human accuracy on our *SFX-Test* and *In-the-Wild Test* datasets.

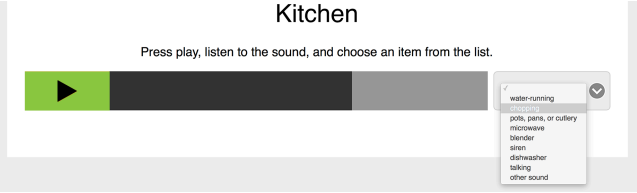

Figure 10. Screenshot of web-based interface used in our human-accuracy baseline studies.

First, we ran a crowd-sourced study on Amazon Mechanical Turk. The crowd interface noted the context (*e.g.*, office) and allowed users to play (and replay) a single sound (Figure 10). Given these two pieces of information, the task was to select the best label (*e.g.*, telephone ringing) from a dropdown list of classes found in that context. Participants could also choose "unknown" if they felt none of the options were correct. In one round of labeling, each of our classes appeared once, plus seven out-of-context ("unknown") sounds (one injected for each context). 250 crowdworkers completed three labeling rounds on *SFX-Test* (producing 27,750 labels), and another 250 crowdworkers labeled our *In-the-Wild Test* set (producing 21,750 labels; less because our real-world data omitted 8 classes, *e.g.*, baby crying).

A potential danger in online studies is reduced accuracy from malicious or apathetic crowdworkers. Thus, as an additional human benchmark, we ran a monitored, in-lab study. This used the same interface and followed the same procedure as the crowd study, but collected four rounds of data instead of three. In total, 50 participants labeled 7,400 sounds from *SFX-Test*, and another 50 participants labeled 5,800 sounds from *In-the-Wild Test*.

Across all contexts, the average accuracy on our *SFX-Test* data was 83.6% (SD=5.9%) for our crowd workers, and 82.7% (SD=7.0%) for our in-lab participants. For our *In-the-Wild Test* set, the accuracy was 83.9% (SD=6.6%) for our crowdworkers and 87.0% (SD=6.1%) for our in-lab labelers. For reference, under comparable test conditions, our system achieves 89.8% and 80.4% accuracy on *SFX-Test* and *In-the-Wild Test* respectively (Figure 9), which is very close to human performance (no significant difference). See also Figure 13 for a breakdown of human accuracy across contexts.

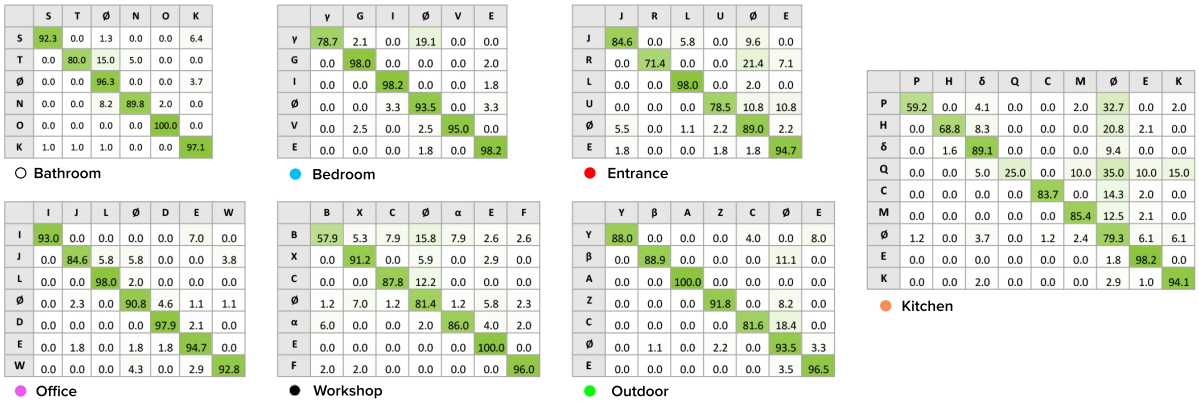

Figure 11. Confusion matrices for *SFX-Test* with 20% unknown classes. Class letter legend found in Figure 7 (Ø is unknown class).

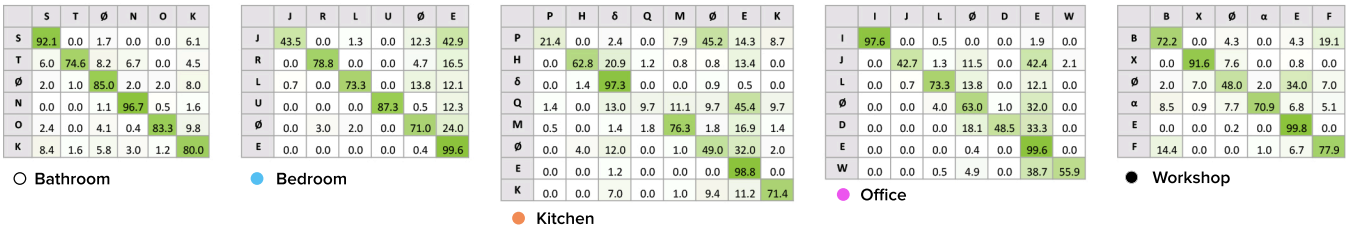

Figure 12. Confusion matrices for *In-the-Wild Test* with 20% unknown classes. Class legend found in Figure 7 (Ø is unknown class).

Location Context

We tested Ubicoustics' performance across seven location contexts (Figure 13), which ranged from 77.4% in the kitchen to 93.9% in the bedroom. Model performance roughly correlates with human performance (R=0.63).

Limiting classes to a context is only possible if a device knows its location. In the case of a smart speaker, a user could specify a location during setup, but a smartwatch is rarely stationary. Thus, we also evaluated our model's ability to automatically infer its physical context (*e.g.*, kitchen *vs.* office). Such a capability could enable devices to automatically load per-context classifiers without user intervention. For this, we used the predicted sound class itself as a proxy for the origin context. For example, if Ubicoustics predicts a *microwave* event, we can infer that the device is in a kitchen.

To simulate this experimentally, we passed our 30-class *SFX-All* model ten random clips for each context in our test data (*SFX-Test* and *In-the-Wild Test*). The model classifies these clips individually, and the output is used to cast a vote for a context. A few classes have special voting logic: if *water running* is detected, votes for both *bathroom* and *kitchen* are cast, and similarly, *knock* casts votes for both *office* and *entrance*. There are also a set of context-free classes (*i.e.*, can happen anywhere) that do not cast any votes (dog bark, cat meow, vacuum, speech, phone ringing, laugh, cough, door, baby cry and hazard alarm). Once all ten sounds have been processed, the context with the highest vote count is chosen and validated against the ground truth context. We repeated this process 1000 times with random contexts and sounds within that context.

Overall, automatic context recognition accuracy is 99.4% (SD=1.5%) for the *SFX-Test* dataset and 92.2% (SD=14.4%) for the *In-the-Wild* dataset. These preliminary results show that it may be possible for interactive systems to automatically deduce their context of use by listening for a short period after setup.

Efficacy of Augmentations

The results reported thus far are based on models tuned on *SFX-All*, which is the superset of all of our data augmentations. To investigate the effects of each augmentation type, we ran the same procedure as our main accuracy studies (20% unknown test sounds, checkpointed on self) but with different tuning sets: *SFX-Orig* (*i.e.*, no augmentations), *SFX-Amp* (amplification), *SFX-Persist* (persistence of sound), *SFX-Mix* (mixing), and *SFX-All* (all augmentations).

In this experiment, *SFX-Orig* serves as a baseline. Combining results from *SFX-Test* and *In-The-Wild Test*, the average delta over the baseline for *SFX-Amp* is +1.6%, *SFX-Persist* is 0.0%, *SFX-Mix* is +1.8%, and *SFX-All* is +2.4%. All but *SFX-Persist* are a significant improvement over baseline (paired t-tests, $p<.01$). See Figures 8 and 9 for a breakdown.

Performance Across Platforms

If we break out the results by device (*In-the-Wild Test* plus 20% unknown test sounds), we can see that the laptop performed the best at 86.1% accuracy, followed by the watch at 84.1% (Figure 14). It appears that better quality microphones and being closer to events helps recognition. The IoT sensor performed the worst at 71.3%, likely because it was often the farthest sensor from the event source. We also saw a performance drop between phone-on-table at 81.5% *vs.* phone-in-pocket at 76.1% (which "muffled" the microphone and often added fabric chafing background noise).

Comparison to Prior Results

There have been a wide variety of metrics used to evaluate sound-based recognition systems that make apples-to-apples comparisons challenging. Here, we discuss baselines that are most relevant to our work.

SoundNet [4] benchmarks its results on the DCASE challenge [39], where it achieves 88% on 10 classes. On the ESC-50 (50 classes) and ESC-10 (10 classes) [29] datasets, SoundNet reports an accuracy of 74.2% and 92.2% respectively. For reference, we achieve an accuracy of 82.1% on 30 classes (all contexts classifier) in our *SFX-Test* set.

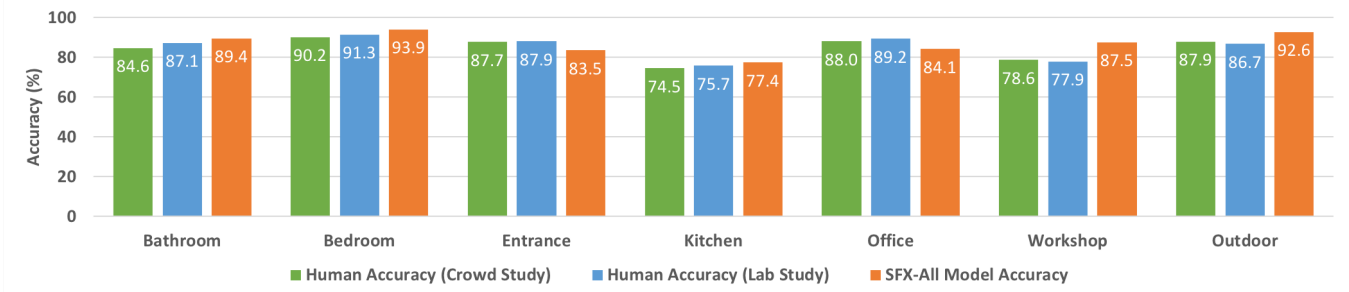

Figure 13. Per-context accuracy (trained on *SFX-All*, tested on *SFX-Test* & *In-The-Wild Test* combined, plus 20% unknown classes.

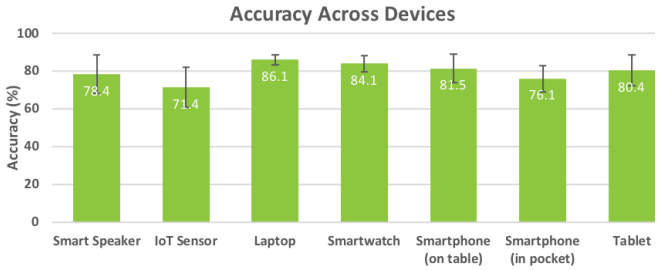

Figure 14. Per-device accuracies on the In-the-Wild Test.

Other systems employ metrics that are more relevant to audio indexing – given sound or video, a system produces prediction labels as metadata to facilitate search. Hershey *et al.* [16] report a best-case mAP (mean Average Precision) of 0.38 on the YouTube-8M dataset and 0.31 on the AudioSet dataset. Stated differently, for every sound clip broken into *n* smaller instances, the system makes the correct prediction roughly every third instance. While they benchmark their system as a retrieval system, we evaluate ours as a recognition system.

Finally, in the activity recognition domain, BodyScope [42] used a wearable necklace to achieve a recognition accuracy of 71.5% (tested on in-the-wild data) across four activities (eating, drinking, speaking and laughing). Similarly, SoundSense [22] recognizes three classes (speech, music and ambient sound) with an accuracy of ~84%. We provide comparable accuracies, while offering a much richer set of activities without requiring any in situ training.

LIMITATIONS AND FUTURE WORK

Our evaluations show promising results that could make sound-based activity recognition more practical. That said, our work also has limitations, which we now discuss along with directions for future work.

Accuracy

Our system achieves an average per-context accuracy of 80.4% (on *In-the-Wild Test* data, checkpointed on independent data, with 20% unknown sounds injected), meaning that roughly one in five sounds is missed or misclassified. This performance is not sufficient to support end-user applications, though we note it is competitive with human accuracy. Better quality microphones and higher sampling rates could certainly increase accuracy. Likewise, it is also possible to leverage better and deeper model architectures, such as ResNets [44] and those that can model audio temporalities such as LSTMs and CRNNs [14].

Simultaneous Events

The real world is often noisy, with multiple sounds occurring simultaneously. However, our current system and experimentation chiefly focused on isolated sounds. No doubt in more chaotic environments, accuracy would suffer. Fortunately, as noted earlier, the additive nature of sound (and sound effect data) is perfect for generating compound sounds, while retaining the benefits of tight segmentation and good labels – an exploration we leave to future work.

Privacy

The richness of sound is a double-edged sword. On one hand, it enables fine grained activity sensing, while also capturing potentially sensitive audio, including spoken content. This is an inherent and unavoidable danger of using microphones as sensors. However, we note that always-listening devices – especially smartphones and smart speakers – are becoming more prevalent and accepted in homes and workplaces, and so the social stigma of such devices *may* wane in the coming years. In the meantime, to mitigate this technically (socially is more challenging), we convert all live audio data into low-resolution Mel spectrograms (64 bins), discarding phase data. With this signal, our model can readily detect speech, but the spoken content is challenging to recover. Moreover, we envision our model being run locally on devices (as we showed possible with our laptop, smartphone, and IoT device), such that audio data never has to be transmitted.

Sound Effect Library Licensing

We were fortunate that our institution had licenses to three of the four sound effect libraries we employed (FreeSound is freely available online). Such subscriptions and licensing are a hindrance to researchers looking to leverage sound effects in future work, as well as replicate studies. For this reason, we have sought the appropriate permissions to release the processed datasets used in this paper (free for education and research purposes).

Bootstrapping Complementary Sensing Systems

Dimensions beyond audio (*e.g.*, vibrations, motion) are useful for digitizing physical events in environments (see *e.g.*, [20]). However, data collection, segmentation and labeling are generally laborious. With Ubicoustics, we can facilitate and bootstrap this process. For instance, in a wearable application, researchers can collect accelerometer data in tandem with audio. Ubicoustics can provide predictions for performed events (*e.g.*, typing, chopping, writing), as well as offering automatic segmentation of data. We hope to explore this multimodal bootstrapping in future work.

EXAMPLE APPLICATIONS

We conclude with several example uses that demonstrate the potential of our system (Figures 15–20 and Video Figure), which span a range of contexts and hardware platforms.

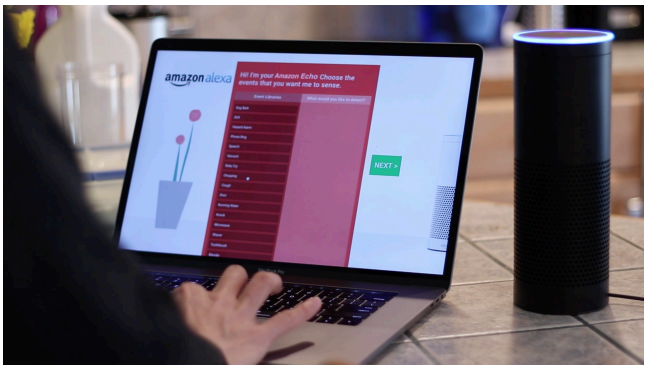

Figure 15. A smart speaker could be configured to pay attention to certain events via a drag-and-drop interface.

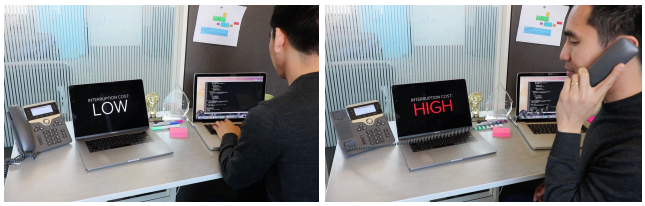

Figure 16. Devices could be made aware of user interruptibility, enabling more nuanced notification behaviors. When working alone, interruption cost might be low (left), whereas in a phone call it might be high (right).

Context-Aware Assistants

Despite smart speakers like Amazon Alexa and Google Home being integrated into people's living spaces (*e.g.*, kitchen, living room, bathroom), these systems have no understanding of events happening around them. With Ubicoustics, we can enable new interactive opportunities that leverage real-time context-awareness. There are two main categories. First are *implicit* interactions, where systems can proactively provide users with assistive information. For example, a system could alert users when someone knocks on their front door or automatically move to the next step in a recipe after detecting, *e.g.*, chopping or a blender running for a defined period. Interactions can also be *explicit*, where users ask their smart assistants about physical events (Figure 15), for example, "is my *microwave done defrosting*?" Notifications about physical events are also possible, such as, *"send me an alert when my laundry is done."* Additionally, knowledge of active tasks can suggest a user's interruptibility and better manage interruptions (Figure 16).

Informatics

Simply logging the occurrence of events could also be valuable. For example, Ubicoustics could track a user's typing over the course of a day, prompting breaks. It could also track the ratio between typing and talking, encouraging more face-to-face interactions. In a classroom setting, a tablet or laptop could track the ratio between teacher and student speech (Figure 17), suggesting better instructional behaviors [6]. Finally, IoT sensors in an industrial setting could track equipment use, helping to schedule maintenance (Figure 18).

Mobile and Wearable Sensing

Smartwatches are unique in that they reside on the body, equipping users with a "sensor" that they carry everywhere they go. As found in our evaluations, watches are one of the stronger performing device categories, given their proximity to hand-triggered events. Interactions with objects could be

Figure18. Sound-driven IoT sensors could track equipment use (A) for safety and maintenance (B).

Figure 17. In a classroom setting (A), tablets could track and visualize instructional informatics, such as speaking ratio (B).

logged for quantified self, safety, and assistive applications (Figure 19). Being proximate to users is also powerful for health sensing. For example, Ubicoustics can detect when a user coughs or sneezes more frequently. This could enable smartwatches to track the onset of symptoms and potentially nudge users towards healthy behaviors, such as washing hands or scheduling a doctor's appointment (Figure 20).

CONCLUSION

We have shown that Ubicoustics can unlock real-time activity recognition by leveraging one of the most common sensors found in consumer electronics today – microphones – bringing the promise of smart devices and environments closer to reality. By leveraging existing state-of-the-art sound classification models and tuning them with sound effects, we enabled a general-purpose and flexible sound recognition pipeline that requires no in situ data collection, yielding a "plug and play" end user experience. We evaluated the robustness of our approach across different physical contexts and hardware platforms, and show that our system can achieve superior accuracies over prior work, both in terms of recognition accuracy and false positive rejection.

ACKNOWLEDGEMENTS

This research was generously supported with funding from the Google Ph.D. Fellowship, Packard Foundation, Sloan Foundation, and Qualcomm. We would also like to thank Ruslan Salakhutdinov, Florian Metze, and Bhiksha Raj for their advice and feedback.

Figure 19. Phones could be used to detect appliance use, which can proactively provide information (Left: "don't forget to wear safety googles!") or launch complementary applications (Right: "3100 RPM for softwood").

Figure 20. Smartwatches could track additional health information, such as coughing (A), and recommend actions (B).

REFERENCES

1. R.E Abbott and S.C. Hadden. 1990. Product Specification for a Nonintrusive Appliance Load Monitoring System. *EPRI Report #NI-101*, 1990.
2. Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. 2016. Youtube-8M: A large-scale video classification benchmark. https://arxiv.org/abs/1609.08675
3. Apple, Inc. Impulse Response Utility, User Manual. 2009. https://documentation.apple.com/en/impulseresponseutility/usermanual, Retrieved on July 12, 2018.
4. Yusuf Aytar, Carl Vondrick, and Antonio Torralba. 2016. SoundNet: Learning sound representations from unlabeled video. *Advances in Neural Information Processing Systems (NIPS '16)*.
5. BBC Sound Effects. https://shop.prosoundeffects.com/products/bbc-complete-sound-effects-library, Retrieved on April 4, 2018.
6. S. D. Brookfield and S. Preskill. 1999. Discussion as a way of teaching: Tools and techniques for democratic classrooms. San Francisco: Jossey-Bass.
7. Peter Doyle. Echo and Reverb: Fabricating Space in Popular Music Recording. 2005. Wesleyan. ISBN 978-0819567949.
8. Benjamin Elizalde, Rohan Badlani, Ankit Shah, Anurag Kumar, Bhiksha Raj. 2017. Never-Ending Learner of Sounds. In *NIPS Workshop on Machine Learning for Audio*, 2017. https://arxiv.org/pdf/1801.05544
9. Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T. Freeman, Michael Rubinstein. 2018. Looking to Listen at the Cocktail Party: A Speaker-Independent Audio-Visual Model for Speech Separation. https://arxiv.org/abs/1804.03619
10. Antti J.Eronen, Vesa T. Peltonen, Juha T. Tuomi, Anssi P. Klapuri, Seppo Fagerlund, Timo Sorsa, Gaëtan Lorho, and Jyri Huopaniemi. 2006. Audio-based context recognition. In *IEEE Transactions on Audio, Speech, and Language Processing* 14, no. 1 (2006): 321-329.
11. Pasquale Foggia, Nicolai Petkov, Alessia Saggese, Nicola Strisciuglio, and Mario Vento. 2015. Reliable detection of audio events in highly noisy environments. In *Pattern Recognition Letters 65* (2015): 22-28.
12. Jon E. Froehlich, Eric Larson, Tim Campbell, Conor Haggerty, James Fogarty, and Shwetak N. Patel. 2009. HydroSense: infrastructure-mediated single-point sensing of whole-home water activity. In *Proc. of the 11th international conference on Ubiquitous computing* (UbiComp '09). ACM, New York, NY, USA, 235-244. DOI=http://dx.doi.org/10.1145/1620545.1620581
13. Jort F. Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio Set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP), pp. 776-780. IEEE.
14. Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. Deep learning. Vol. 1. Cambridge: MIT Press.
15. Sidhant Gupta, Matthew S. Reynolds, and Shwetak N. Patel. 2010. ElectriSense: single-point sensing using EMI for electrical event detection and classification in the home. In *Proceedings of the 12th ACM international conference on Ubiquitous computing* (UbiComp '10). ACM, New York, NY, USA, 139-148. DOI: https://doi.org/10.1145/1864349.1864375
16. Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal et al. 2017. CNN architectures for large-scale audio classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP), 2017, pp. 131-135. IEEE.
17. Mike Lam, Mostafa Mirshekari, Shijia Pan, Pei Zhang, and Hae Young Noh. 2016. Robust occupant detection through step-induced floor vibration by incorporating structural characteristics. In *Dynamics of Coupled Structures*, Volume 4, pp. 357-367. Springer, Cham.
18. Nicholas D. Lane, Petko Georgiev, and Lorena Qendro. 2015. DeepEar: robust smartphone audio sensing in unconstrained acoustic environments using deep learning. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)*. ACM, New York, NY, USA, 283-294. DOI: https://doi.org/10.1145/2750858.2804262
19. 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. DOI: https://doi.org/10.1145/2702123.2702416
20. 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. DOI: https://doi.org/10.1145/3025453.3025773
21. 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. DOI: https://doi.org/10.1145/2702123.2702178

22. 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 *Proceedings of the 7th international conference on Mobile systems, applications, and services* (MobiSys '09). ACM, New York, NY, USA, 165-178. DOI=http://dx.doi.org/10.1145/1555816.1555834
23. Brian McFee, Eric J. Humphrey, and Juan P. Bello. 2015. A Software Framework for Musical Data Augmentation." In *Proceeding International Society for Music Information Retrieval Conference (*ISMIR 2015), pp. 248-254.
24. Ian McLoughlin, Haomin Zhang, Zhipeng Xie, Yan Song, and Wei Xiao. 2015. Robust sound event classification using deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* 23, no. 3 (2015): 540-552.
25. Vinod Nair, and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. 2010. In *Proceedings of the 27th international conference on machine learning* (ICML-10).
26. Network Sound Effects. https://www.sound-ideas.com/Product/199/Network-Sound-Effects-Library, Retrieved on July 12, 2018.
27. Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen. 2017. Convolutional recurrent neural networks for polyphonic sound event detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25, 6: 1291-1303. DOI: https://doi.org/10.1109/TASLP.2017.2690575
28. Huy Phan, Lars Hertel, Marco Maass, and Alfred Mertins. 2016. Robust audio event recognition with 1-max pooling convolutional neural networks. https://arxiv.org/abs/1604.06338
29. Karol J. Piczak. 2015. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd ACM international conference on Multimedia* (MM '15). ACM, New York, NY, USA, 1015-1018. DOI: https://doi.org/10.1145/2733373.2806390.
30. Jay Rose. 2008. Producing Great Sound for Film and Video. 3rd Edition. Focal Press and Elsevier Inc. ISBN 978-0-240-80970-0.
31. Thomas D. Rossing, F. Richard Moore, and Paul A. Wheeler. 2001. The Science of Sound. 3rd Edition. Pearson. ISBN 978-0805385656.
32. Justin Salamon and Juan Pablo Bello. 2017. Deep convolutional neural networks and data augmentation for environmental sound classification. In *IEEE Signal Processing Letters.* 24.3 (2017): 279-283.
33. Justin Salamon and Juan Pablo Bello. 2015. Feature learning with deep scattering for urban sound analysis. In *Proc. 23rd European Signal Processing Conference* (EUSIPCO), 2015. IEEE.
34. Seeed Studio. ReSpeaker 2-Mic PHAT. https://www.seeedstudio.com/ReSpeaker-2-Mics-Pi-HAT-p-2874.html, Retrieved on July 18, 2018
35. ShotSpotter. www.shotspotter.com, Retrieved on April 3, 2018.
36. Soundsnap. https://www.soundsnap.com, Retrieved on July 12, 2018.
37. Johannes A. Stork, Luciano Spinello, Jens Silva, and Kai O. Arras. 2012. Audio-based human activity recognition using non-markovian ensemble voting. In *RO-MAN*, 2012 IEEE, pp. 509-514. IEEE.
38. Johannes. A. Stork, L. Spinello, J. Silva and K. O. Arras, "Audio-based human activity recognition using Non-Markovian Ensemble Voting," *The 21st IEEE International Symposium on Robot and Human Interactive Communication* (RO-MAN 2012), 509-514. DOI: 10.1109/ROMAN.2012.6343802
39. Dan Stowell, Dimitrios Giannoulis, Emmanouil Benetos, Mathieu Lagrange, and Mark D. Plumbley. 2015. Detection and classification of acoustic scenes and events. In *IEEE Transactions on Multimedia*, 17, no. 10 (2015): 1733-1746.
40. Tanzeem Choudhury, Sunny Consolvo, Beverly Harrison, Jeffrey Hightower, Anthony LaMarca, Louis LeGrand, Ali Rahimi et al. 2008. The mobile sensing platform: An embedded activity recognition system. In *IEEE Pervasive Computing*, vol. 7, no. 2, pp. 32-41, April-June 2008. DOI: 10.1109/MPRV.2008.39
41. Jamie A. Ward, Paul Lukowicz, Gerhard Troster, and Thad E. Starner. 2006. Activity recognition of assembly tasks using body-worn microphones and accelerometers. In *IEEE transactions on pattern analysis and machine intelligence*, 28, no. 10: 1553-1567.
42. Koji Yatani and Khai N. Truong. 2012. BodyScope: a wearable acoustic sensor for activity recognition. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (UbiComp '12). ACM, New York, NY, USA, 341-350. DOI=http://dx.doi.org/10.1145/2370216.2370269
43. Frederic Font, Gerard Roma, and Xavier Serra. 2013. Freesound technical demo. In *Proceedings of the 21st ACM international conference on Multimedia* (MM '13). ACM, New York, NY, USA, 411-412. DOI: https://doi.org/10.1145/2502081.2502245
44. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.