Dumbo: interactive machine learning for activity recognition from ambient sound

Albert Mwanjesa
University of Utrecht
Utrecht, the Netherlands
a.j.mwanjesa@students.uu.nl

Shiqi Bai
University of Utrecht
Utrecht, the Netherlands
s.bai@students.uu.nl

Irina Bianca Serban
Eindhoven University of Technology
Eindhoven, the Netherlands
irina.b.serban@gmail.com

Mathias Funk
Eindhoven University of Technology
Eindhoven, the Netherlands
m.funk@tue.nl

CHI'20,, April 25–30, 2020, Honolulu, HI, USA
ACM 978-1-4503-6819-3/20/04.
https://doi.org/10.1145/3334480.XXXXXXX

Abstract
Environmental sound recognition [2] in domestic settings is crucial for smart products and automation as ambient sound contains rich information about activities. Previous approaches employ machine learning, yet, domestic contexts are highly diverse. End-users' personal activity labels have not been studied, nor incorporated in activity recognition. This case study shows the design process of Dumbo, an interactive device designed to recognize and label human activities in a domestic environment from ambient sound. The design employs transfer learning (TL) [10] on live audio signals and allows for end-user labeling and relabeling on a Raspberry Pi. Dumbo operates locally to protect privacy. We present user interface and hardware design as well as different approaches to training the initial and final models. We show that TL outperforms previous approaches with a fully pre-trained model and we conclude with a reflection on how interactive machine learning was integrated into the design process and which challenges have to be tackled to generalize this approach.

Author Keywords
Activity Recognition; Ambient Sound; Transfer Learning; Interactive Machine Learning; Sensing System; Tangible Interaction; Personalization

CCS Concepts

•Computing methodologies → **Transfer learning; Neural networks; •Human-centered computing** → *User interface design; Ethnographic studies;*

Introduction

Music classification, automatic speech recognition or biometric voice authentication are some of the forms through which sound recognition has achieved commercial success in nowadays volume production. However, there is a new "player" looming in the field of the Internet of Things (IoT) —automatic environmental sound recognition (AESR) [2]. The sounds a person makes in their own home is a rich source of information for an IoT smart home system. With the help of this data, the ongoing human activities can be detected and used as input for different purposes such as security devices or the adaptation of lighting and heating systems. However, while the aforementioned sound classifications focus on a smaller range of sounds with more specific features, ambient sound recognition poses many problems. Because of these problems, the explorations in the AESR niche have been reduced more to the research field for now. The first problem in AESR is one of the **acoustic features**. Compared to speech which has grammar and music which has score and rhythm, the ambient sound does not have any specific attributes; moreover, it can also have background noise and it has no pattern since it can be followed by any random noise. Secondly, ambient sound labeling implies constant listening by the system; uploading the entire day sound snippet to the cloud can be considered as privacy-invading, an **ethical** problem. On the other hand, detecting and processing sound locally 24/7 requires precise usage of **computational** resources to ensure responsiveness. This second problem is a dilemma between privacy and responsiveness. Finally, different homes have different acoustics and, therefore, certain activities can sound different in distinct contexts; moreover, the device needs to be **contextually** aware which constitutes a problem when it comes to mobile phone microphones (they can be covered in certain scenarios - when placed in a bag or a pocket); additionally, one sound might be labeled as two different activities for two different home environments because of reasons such as culture; last but not least, the initial ontology on which the system was trained can be limited in such a way that it does not cover or match the variety of activities and rituals a certain subject performs in the home —a boxing enthusiast skipping rope in their living room might sound the same as a circus trainer practicing their whipping skills to the machine.

Consequently, we posed the question: what if the user could change the ontology and influence the system's labeling mechanism based on their rituals and their home environment? Moreover, we saw a challenge of creating a system that could, on low computational power, provide real-time activity detection without uploading the data of the user on the Cloud, therefore, solving the problem of intrusiveness.

Related work

When looking at existing work, ambient sound recognition is a popular research topic. However, most of the explorations have been focusing on evaluations of accurately labeling sounds in different contexts (public places, home, car, bathroom, etc.) or of runtime and CPU usage. For example, CondioSense implements an active audio sensing method on commodity mobile phones and evaluates the accuracy of the detection given different contexts of the phone [9]. AmbientSense, on the other hand, evaluates the efficiency of real-time ambient sound detection in "autonomous mode" (locally labeling the sounds) vs. in "server mode" (with the support of a server) in terms of running time, recognition

Dumbo —functions and user capabilities

Functions: signaling it is listening, signaling it has detected and labeled an activity, communicating the current detected activity.

User capabilities: accepting or denying a label, relabeling an activity - with an existing label or by giving it a new name (in case the device has made an error), listening to activity snippets from the current day, reviewing a daily list of detected and labeled activities, modifying a list of labels (re-naming or adding labels).

Activity = a sound snippet recognized and labeled by the system

Label = a name given to a group of activities; "Shower" is the label of all the sound snippets recorded while the user was showering

time and CPU usage [11]. Another related piece of work is UbiCoustics, a system that uses a commodity microphone to accurately label human activities by focusing on how well sound augmentation improves the performance of the model [8].

Design requirements and challenges

However, we decided to approach the problem from a different angle by materializing AESR in a more user-centered manner. Therefore, we designed Dumbo, a device that is specifically built to detect and label human activities in the home environment by analyzing ambient sounds. On top of this, Dumbo has a user interface (UI) which allows the user to give feedback to the system to personalize classification depending on the home. Our challenge was represented by designing a system which (1) performs real-time operation on contextual ambient sound, (2) processes sounds locally without uploading the content to the Cloud, (3) labels the sounds with the help of Machine Learning to the corresponding human activities, and (4) receives user feedback (accepting label/denying label/renaming label/adding label to ontology) to allow for personalising of the labeling system for each home environment.

While previous work focused more on the high performance of AESR either locally or in a server-supported system, in different contexts and with sound-augmentation, Dumbo wants to explore the possibility of an AESR model which can "mold" its detection algorithm based on distinct home environments without using a server-based application. With this goal in mind, certain challenges had to be solved: first, to eliminate the problem of the mobile phone microphone's context, Dumbo is a physical device that uses a commodity microphone. Second, to allow for smooth detection on low computational power, Dumbo performs on a Raspberry Pi 3 model b+; in this way, classification, retraining, and processing user feedback are done locally. Third, to allow for a more efficient re-training every time user feedback is received, Dumbo uses Transfer Learning (TL) [10], a method in machine learning in which one model is fine-tuned using data from a different distribution; in this case, an activity recognition model was fine-tuned using the user's audio data. Finally, the user interaction is designed into the physical device with limited features for privacy issues to allow for system feedback and user input, while more user input can be processed through a web application that communicates with the Raspberry Pi in real-time.

Further, we present the design of the UI and hardware components, followed by the process and validation of interactive machine learning implementation. Results from two types of evaluation (TL iterations and an informal user test) are presented with avenues for future work. We conclude with an assessment of DUMBO as a proposition of a customizable AESR device related to the use of interactive machine learning in the domestic context.

User interface design

The design of the UI was only constrained by the above-mentioned design requirements. However, clear limitations existed in terms of strictly-local operation, computational resources, and hardware space constraints. We addressed this challenge in an iterative process that combined the design of the web application and the physical product that would enclose also the machine learning components to drive the core functionality. This design process was carried out in two phases: content strategy and interaction design.

Content strategy

The rationale behind the type of device and the content of the product was designed based on the functions and user capabilities. An exploration of forms of UI according to ex-

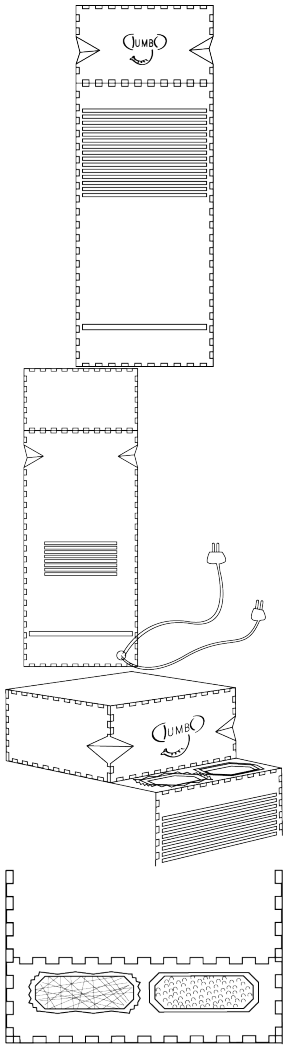

Figure 1: Final sketches of the physical prototype

amples from the industry of smart systems was conducted. The functions and capabilities needed to be introduced in the design either as a single product or as a system with two parts: a main locus of interaction and a remote. The first exploration consisted of a single device with only a touchscreen. The second exploration was a physical UI (with limited interaction) and a touchscreen graphical UI (web application). The third exploration was a single device with a touchscreen and physical controls for the main functions of the touchscreen.
The second form (physical UI + web application) was finally chosen as it offered an opportunity for designing an intuitive, yet exciting user experience within the physical device, a higher level of privacy for the user's data within the app (smartphone is a personal device), a more innovative design compared to a simple touch screen, and a more versatile, mobile system. The physical device can be moved to a location with good sound properties, while the screen remains ubiquitously accessible from the mobile phone.

The functionalities of the physical device versus the digital interface were divided: the physical device would signal listening, detected activities, and the current label (with the highest confidence). With the help of the mobile application, the same functionalities would be accessible on-demand, including changing a denied label, listening to daily activity snippets, reviewing a daily list of activities and modifying a list of existing labels (add or rename).

In this way, the physical device engages the peripheral attention [1] (listening and detecting an activity), reducing the need for continued attention, yet allowing for in-the-moment interaction. In contrast, the web application presents more information (i.e., the daily overview and list of labels) and allows for personalization and remote control of the system.

Interaction design
The interaction design of the physical DUMBO had to support end-users in two main tasks which were mapped to the corresponding interactions. Firstly, Dumbo had to **listen and detect**. To operate autonomously over a longer period without the user's attention, listening and detection of activities had to be signaled in a non-intrusive way. We employed an LED strip with three states: off if Dumbo would be off, constantly on if Dumbo would be listening, and animated if Dumbo would have detected an activity. Movement instead of color was used for the LEDs for the engagement of the user's perceptual skills [4]. Secondly, the user needed to be able to **inquire and approve of the current labels** at any time. Inquiring was implemented with an open-close functionality on the hardware device. When the user shifts the top part, the current label is presented through text-to-speech together with options to accept or reject the labeling. This design decision was motivated by metaphorically revealing a secret to the user's central attention [1]. Only with this attention, the device presents an interface, making mode-switching physical for a less confusing interaction [5]. Finally, we solve the problem of activity interruption: the open state means that the currently performed activity was interrupted and only after closing it might resume (see Fig. 1).
Next to the physical device, a web application with more extensive functionality was developed, which allows users to view the current predicted activities, to provide feedback on these predictions (i.e., listening to the recorded sound again and accept or reject predicted labels), to manage activity labels in a daily timeline of activities. The web application extends the physical device throughout the living space.

Hardware design

The possibilities for the shape of Dumbo (spheric, cubic or cylinder), type of open-close mechanism (sliding, hinge,

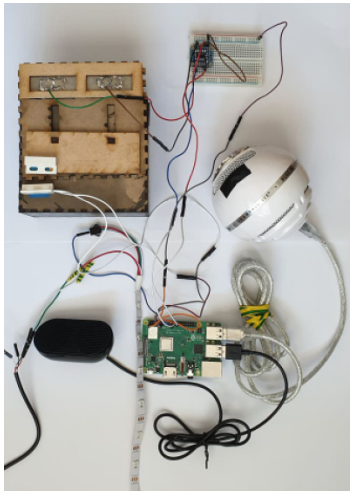

Figure 2: Physical DUMBO hardware wiring

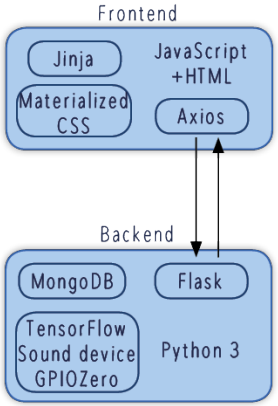

Figure 3: Backend and front-end architecture of Dumbo (both physical and digital)

extendable, fan), type of material texture (harsh - laser engraved felt, soft - suede, in-between harsh and soft - felt, smooth - latex) and the type of interaction for denying/accepting the label (sliding, tapping, pushing, pulling, stroking) seemed endless. We approached this with an online survey (N=42) followed by a focus group with paper prototypes (N=3). There was a slight preference for the cubic shape. We chose a Raspberry Pi 3B+, together with a USB speaker (for communicating the label and confirming input), a USB microphone, a magnetic contact switch (sliding mechanism), digital LEDs (activity indication) and a standalone 5-pad capacitive sensor attached to a breadboard (conductive fabric for accept and reject controls; see Fig. 2). The case of the Dumbo was laser cut and we added handles to guide the fingers, according to focus group feedback. The functionalities of the switches and sensors were programmed in Python (GPIOZero library) (see Fig. 3).

Interactive machine learning

The goals of Dumbo were to train and deploy a (partially) retrainable model that can classify domestic sounds reliably. To find the most appropriate machine learning strategy, several approaches were tested. The underlying challenge was to replicate the target context with appropriate audio data to allow for testing and validation.

Initial training with AudioSet

First, AudioSet [6] was used to train to the model using architecture and related parameters (e.g., learning rate) of the VGGish model [7]. During the first trials, both balanced and unbalanced training were implemented: unbalanced training directly used the whole AudioSet to train the model while for balanced training, the AudioSet data was resampled so that the number of samples for each class remained the same. The two models that were trained on unbalanced data and balanced data from AudioSet led to unsatisfying results when tested on a validation dataset (obtained by self-recording activities in the home). The two training approaches returned models with accuracies below 10%. These results could be caused by acoustic differences between recording contexts: self-acquired contextual audio data were recorded with smartphones whereas the AudioSet data was directly extracted from thousands of YouTube videos.

Data augmentation

In a second approach, we built our own domestic sound database and sound ontology composed of a list of common activities in the domestic environments. Each ontology concept matched the recorded sound clips. To further increase the quantity of the data, subtle audio processing in the shape of frequency-filtering was employed to vary the auditory data. Initially, a bandpass filter for frequency was used on 1483 snippets (i.e ten-second audio snippets). As a result, 37015 auditory snippets were obtained for each sample rate (i.e 11025Hz and 22050HZ). This approach, however, did not provide the needed improvement. The classifier resulting from training the enlarged dataset did not provide better results than the first approach. Eventually, the classification result on unseen data was less than 20%.

The underlying causes why the performance of the models was so low were apparent —1) the differences of acoustics in AudioSet and 2) the frequency filter drastically varying the recorded data. Given this, the augmentation approach was disregarded. The focus switched to a model that could solve the problem of adapting to different acoustic contexts.

Final approach with transfer learning

Finally, we explored the SINS data set [3] as well as transfer learning (TL) [7]. TL is the method of learning from a well-trained model. The benefit of implementing this kind of approach is that the most representative feature of domes-

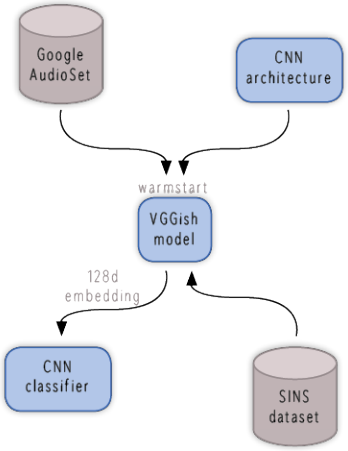

Figure 4: First Transfer Learning iteration

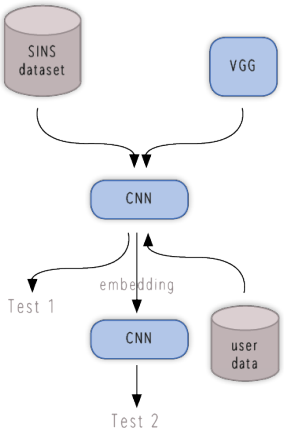

Figure 5: Second Transfer Learning iteration

tic sound can be inherited through a previously well-trained model. This could facilitate forming a new well-tuned model in Dumbo's case [10]. TL was conducted twice, eventually.

The pre-trained model (i.e., VGGish [7]) was used in conjunction with the SINS data to form a new model which is more *familiar* with the acoustics of domestic sounds. In the first step, VGGish is considered the feature extractor and helps extract the most representative features of SINS data, which then would be fed to a new classifier. The new classifier is trained using the extracted features to classify the SINS data (see fig. 4).

Based on the goal of adapting to a new user or environment, transfer learning was applied again. In this step, the real data recorded by Dumbo's microphone was used. A new classifier was added on top of the model obtained with the help of the first iteration of transfer learning. This new classifier was trained using data from the real world, namely the feedback the user has provided by using the app or/and the physical controls (see fig. 5). This step happens iteratively, creating a new classifier every time given user feedback to adapt better to the user's labeling after each retraining step.

Evaluation

The most promising implementation of transfer learning was evaluated to determine its accuracy and how well it would perform under real-life conditions.

Method

The evaluation had two aims: first, to provide Dumbo with a classifier that grasps what sound is and understands the relation between domestic sounds and human activities, and second, to ensure that Dumbo can flexibly adapt to a new environment. The hypothesis for testing was two-fold: (1) the performance of the model from first-time transfer learning would be better than the one trained only on AudioSet when tested on SINS data set, and (2) the model from the second transfer learning would entail higher accuracy compared to the one from first-time transfer learning when tested on real user data. Based on the first hypotheses, the test was performed computationally without access to a real context as the goal was to test whether the model from first-time transfer learning would generally be able to relate human activities and domestic sounds. The second hypothesis, i.e., Dumbo adapts to a new environment, required both a real context and user involvement. As a starting point, Dumbo was equipped with the model from the first transfer learning iteration. The participant used Dumbo in a small apartment for two consecutive days. While Dumbo recorded and classified ambient sounds, the user had to give feedback on the provided classifications.

Findings and discussion

The results of the first objective are presented in table 1. The model created using VGGish and TL with the SINS database outperformed the VGGish model by far when applied to the test data from the SINS database. This successfully proves our first hypothesis. Furthermore, it shows that the model created using TL displays adaptation by gaining a contextual understanding of the feature space it is transferred to.

The second objective of a real-life test was not successful. First, it was difficult to obtain enough audio material for retraining as well as user feedback on the labeling due to constraints and technical issues discussed further. In the given time, the device recorded too few user-provided labels that the model retraining session could not continue. Thus, reliable results to support the second hypothesis were not obtained.

Nevertheless, in its current state, Dumbo forms a platform

Model	Accuracy
AudioSet Model	<10%
AudioSet+SINS TL	92.49%

Table 1: The table shows that, in our case, the model got from the first-time transfer learning can do better than the original AudioSet model in terms of domestic sound classification, the two models were tested only on SINS test data to check the ability of classifying domestic sound.

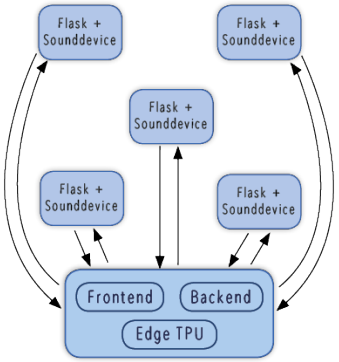

Figure 6: How distributed DUMBO devices could communicate with the central system with the addition of a TPU and the possible use of Tensorflow Lite

that other researchers can build upon. As shown in previous sections, using transfer learning is a promising method for AESR devices to adapt to an individual user's home. However, this team has suggestions on how to improve the user testing method as well as Dumbo itself for future work to investigate the adaptability of the system given user feedback. Conceptually, Dumbo is a device which registers change over time. Therefore, we suggest a longer test session (e.g. 7 days) that could gather more valid user feedback included in the model retraining. Moreover, to boost the feedback, the user should be reminded to check the state of Dumbo (either through app notifications, or sound signals from the physical device when detecting an activity) as the system can be quite ubiquitous. This would help correct the wrong-labeled audio clips so that the adaptability would show improvements.

Moreover, Dumbo encountered problems with predicting new audio signals from the user and storing these signals in larger quantities. The prediction time for a 10-second audio snippet is on average 24 seconds, which makes it difficult for the user to provide timely feedback directly on the device. It is, however, possible to provide feedback on the timeline page of the web application. Luckily, this research team has suggestions for solving both of these problems. A tensor processing unit (TPU) device [1] for the Raspberry PI can accelerate the prediction time, especially combined with the Tensorflow Lite framework[2]. A TPU is a computer chip especially built for machine learning. A conventional CPU chip struggles to run matrix or tensor operations as fast as a TPU. This framework is aimed precisely at solving the problem Dumbo has, namely accelerating the prediction.

In terms of UI and UX design, the user deemed the interface "straight-forward, intuitive and minimalistic". For future development, a status indicator would benefit the user in observing when Dumbo has ran into errors to avoid confusion. However, a variation of the functionalities could also be implemented and tested such as playback from the physical device and not from the app, as the current concept is built. Additionally, Dumbo could be designed into a distributed system (a main locus of interaction and satellite listening devices) to accurately cover the whole domestic environment (see Fig. 6).

[1] https://cloud.google.com/edge-tpu/

[2] https://www.tensorflow.org/lite

Conclusion

In this paper, we describe a case study of integrating AESR in the home in a customizable way with the help of interactive machine learning and user feedback. We show how transfer learning used for the training of a suitable machine learning model is a promising method for an adaptive AESR system. The model delivers good accuracy, yet a first validation failed under real-life conditions. As future steps, we propose a longitudinal deployment of Dumbo as a distributed system and certain hardware and software improvements such as using a TPU in combination with a Tensorflow Lite framework.

Acknowledgements

We thank Ronald Poppe for guidance and helping bring Dumbo to life.

REFERENCES

[1] S. Bakker, E.A.W.H. Hoven, van den, and J.H. Eggen. 2010. Design for the periphery. In *Proceedings of the Eurohaptics 2010 symposium Haptic and Audio-Visual Stimuli : Enhancing Experiences and Interaction, July 7, Amsterdam, The Netherlands (CTIT Workshop Proceedings Series)*, A. Nijholt, E.O. Dijk, P.M.C. Lemmens, and S. Luitjens (Eds.). Universiteit Twente, 71–80.

[2] Sachin Chachada and C.-C. Jay Kuo. 2014. Environmental sound recognition: a survey. *APSIPA Transactions on Signal and Information Processing* 3 (2014), e14. DOI: http://dx.doi.org/10.1017/ATSIP.2014.12

[3] G. Dekkers, S. Lauwereins, B. Thoen, M. W. Adhana, H. Brouckxon, T. van Waterschoot, B. Vanrumste, M. Verhelst, and P. Karsmakers. 2017. The SINS database for detection of daily activities in a home environment using an acoustic sensor network. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*. 32–36.

[4] J.P. Djajadiningrat, S.A.G. Wensveen, J.W. Frens, and C.J. Overbeeke. 2004. Tangible products : redressing the balance between appearance and action. *Personal and Ubiquitous Computing* 8, 5 (2004), 294–309. DOI: http://dx.doi.org/10.1007/s00779-004-0293-8

[5] J.W. Frens. 2006. *Designing for rich interaction : integrating form, interaction, and function*. Ph.D. Dissertation. Department of Industrial Design. DOI: http://dx.doi.org/10.6100/IR608730 Proefschrift.

[6] 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)*. IEEE, 776–780.

[7] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson. 2017. CNN Architectures for Large-Scale Audio Classification. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. https://arxiv.org/abs/1609.09430

[8] Gierad Laput, Karan Ahuja, Mayank Goel, and Chris Harrison. 2018. Ubicoustics: Plug-and-Play Acoustic Activity Recognition. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. ACM, New York, NY, USA, 213–224. DOI: http://dx.doi.org/10.1145/3242587.3242609

[9] Fan Li, Huijie Chen, Xiaoyu Song, Qian Zhang, Youqi Li, and Yu Wang. 2017. CondioSense: High-quality Context-aware Service for Audio Sensing System via Active Sonar. *Personal Ubiquitous Comput.* 21, 1 (Feb. 2017), 17–29. DOI: http://dx.doi.org/10.1007/s00779-016-0981-1

[10] S. J. Pan and Q. Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (Oct 2010), 1345–1359. DOI: http://dx.doi.org/10.1109/TKDE.2009.191

[11] M. Rossi, S. Feese, O. Amft, N. Braune, S. Martis, and G. TrÃűster. 2013. AmbientSense: A real-time ambient sound recognition system for smartphones. In *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. 230–235. DOI: http://dx.doi.org/10.1109/PerComW.2013.6529487