# Find My Things: Personalized Accessibility through Teachable AI for People who are Blind or Low Vision


Linda Yilin Wen
Microsoft Research
linda.wen@microsoft.com

Cecily Morrison
Microsoft Research
cecilym@microsoft.com

Martin Grayson
Microsoft Research
martin.grayson@microsoft.com

Rita Faia Marques
Microsoft Research
t-rimarq@microsoft.com

Daniela Massiceti
Microsoft Research
dmassiceti@microsoft.com

Camilla Longden
Microsoft Research
camilla.longden@microsoft.com

Edward Cutrell
Microsoft Research
cutrell@microsoft.com



## ABSTRACT

The opportunity for artificial intelligence, or AI, to enable accessibility is rapidly growing, but widely impactful applications can be challenging to build given the diversity of user need within and across disability communities. Teachable AI systems give users with disabilities a way to leverage the power of AI to personalize applications for their own specific needs. We demonstrate Find My Things as an end-to-end example of applying Teachable AI systems to address the diversity of accessibility needs. An application that can be taught by people who are blind or low vision to find their personal things, Find My Things illustrates the potential Teachable AI holds for accessibility.



## CCS CONCEPTS

• **Human-centered computing** → Accessibility; Accessibility systems and tools; Accessibility; Accessibility design and evaluation methods.

## KEYWORDS

Accessibility, Artificial Intelligence, Teachable AI

**ACM Reference Format:**
Linda Yilin Wen, Cecily Morrison, Martin Grayson, Rita Faia Marques, Daniela Massiceti, Camilla Longden, and Edward Cutrell. 2024. Find My Things: Personalized Accessibility through Teachable AI for People who are Blind or Low Vision. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3613905.3648641


## 1 INTRODUCTION

The power of artificial intelligence (AI) to enable accessibility is growing and will continue to do so rapidly with the deployment of services based on large foundation models. Despite the opportunity, the diversity of user needs both across and within disability categories can present a challenge to creating broadly usable and efficacious AI systems for accessibility. Further, many machine learning capabilities do not generalize well enough to create compelling, real-world experiences, despite articulated user need demonstrated through heavy usage of apps that provide remote human assistance[1].

Teachable AI systems give users with disabilities a way to leverage the power of AI to personalize applications for their specific needs [7]. They do this by allowing users to teach the AI system about what they need by providing examples to the AI system in a teaching loop (e.g., [13]). In this loop, the user provides a small number of training examples, high-level constraints, or prompts, to train or fine-tune an AI system. The user then receives feedback on system performance through application use, or explanation. Through iteration, the user builds their own mental model of how the AI system works, optimizing it for their own goals.

In this interactivity, we present Find My Things, an example of a Teachable AI system. Designed in conjunction with a citizen design team, Find My Things helps people who are blind or low vision locate their personal items. As shown in Figure 1, users of Find My Things are supported with instructions and auditory / haptic feedback to create four diverse videos of a personal object that they want to teach the AI system to recognize. Within seconds, a personalized AI model is created on device for this personal object. Users can then activate the app to locate and be guided to their personal object with auditory, haptic, and visual cues. Find My Things can be seen as a relatively simple example of the way teachable AI can broaden an AI system – object recognition in this case – to meet the individual needs of a more diverse set of users.

## 2 RELATED LITERATURE

### 2.1 Interactive Machine Learning

Interactive machine learning allows users to iteratively provide data examples and high-level constraints to a machine learning model to continually adapt its performance [2, 13]. The rapid, incremental

[1]Be My Eyes connects people needing sighted support with volunteers and companies through live video around the world. https://www.bemyeyes.com/.



*CHI EA '24, May 11–16, 2024, Honolulu, HI, USA*



ACM ISBN 979-8-4007-0331-7/24/05
https://doi.org/10.1145/3613905.3648641

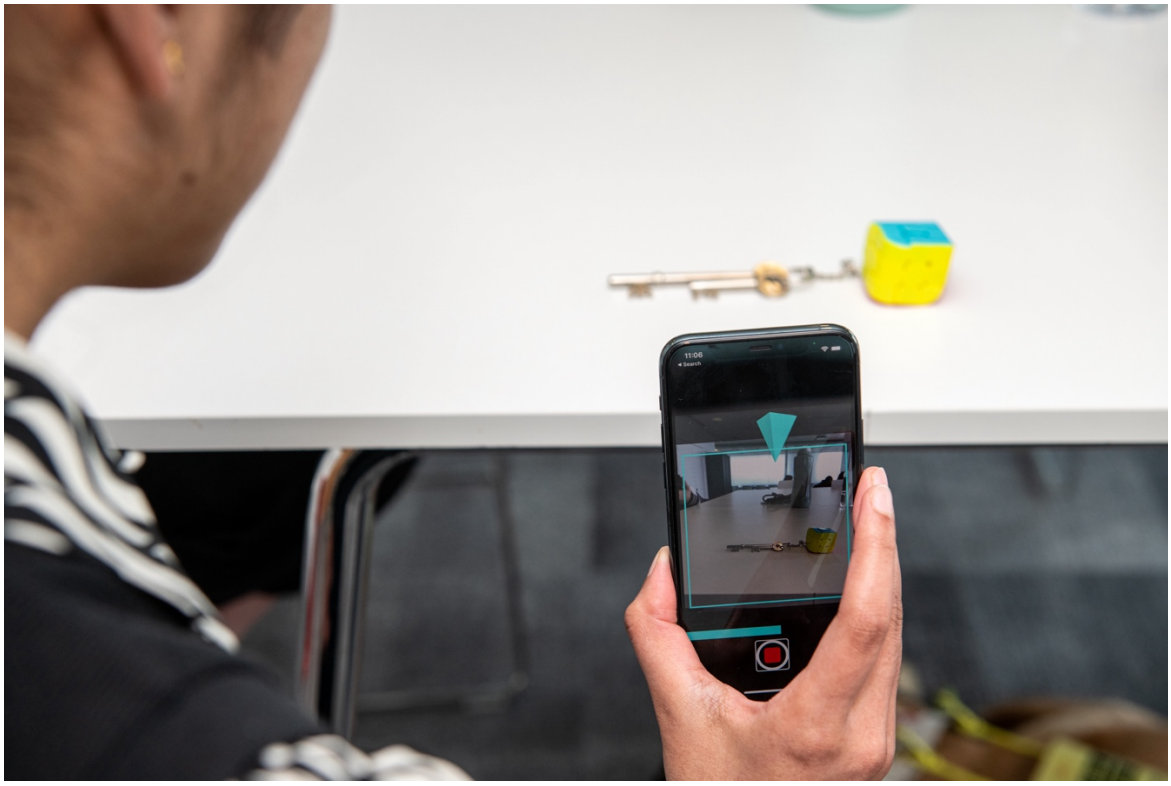

**Figure 1: A user finds a set of keys with Find My Things, having previously taught the keys to the app by providing four videos.**

interaction cycles encourage a close coupling between user and resultant machine learning model. One of the key challenges of interactive machine learning systems is supporting the mental model of the user during the interactive process of refinement. The back and forth between user and ML model to get to the desired result is what we would call a teaching loop. Sanchez et al. [14] proposed several guidelines for designing a teaching loop, such as providing guidance for building teaching sequences and allowing modifications to past teaching actions and sequences of actions.

## 2.2 Teachable AI for Disability

Teachable AI for disability has been proposed as a mechanism to give people with disabilities the agency to personalize experiences to their own needs and situations [7]. It could be adapting previously inaccessible tools or making a new class of tools [9, 12, 17]. Most examples of teachable AI for disability have been teachable object recognizers for people who are blind or low vision, e.g., [1, 5, 8]. Kacorri et al. [8] illustrate that users needed guidance in taking their images, as many used extreme points of view. Follow-up work has explored different strategies to guide the taking of images, such as leveraging ARKit[2], providing sonified and verbal feedback [1], and using hand-to-hand referencing [10].

[2]ARKit is Apple's software development kit that enables app developers to incorporate augmented reality.

## 2.3 Few-Shot Learning

Few-shot learning is an area of machine learning research that aims to reduce the number of examples required to complete a machine learning task, e.g., [16]. This in turn enables AI models to be adapted to diverse, real-world contexts. Adding a new object category to a typical deep learning model would require 100s to 1000s of high-quality labelled examples [18]; in contrast, a few-shot model would require just 5-10 examples. Meta-learning algorithms, which "learn to learn," hold particular promise for interactive applications as they allow for lightweight, adaptable recognition, e.g., [19]. The collection of new datasets such as ORBIT [11] also made it possible for few-shot learning to be applied to real-world challenges. The ORBIT dataset is a collection of videos recorded by people who are blind or low vision on their mobile phones of personal objects that they would like to recognize. The advances in few-shot learning and the publication of the ORBIT dataset provided the foundation for developing the Find My Things app.

# 3 CITIZEN DESIGN TEAM

We brought together a citizen design team of eight blind or low vision young people between the ages of 14 and 25 to collaborate with our research team in the design process of Find My Things. Citizen designers were all young people who had been educated as students with a visual impairment. Our cohort consisted of three braillists and five print users, using screen reader technology and magnification respectively, to access their phones. The goal of

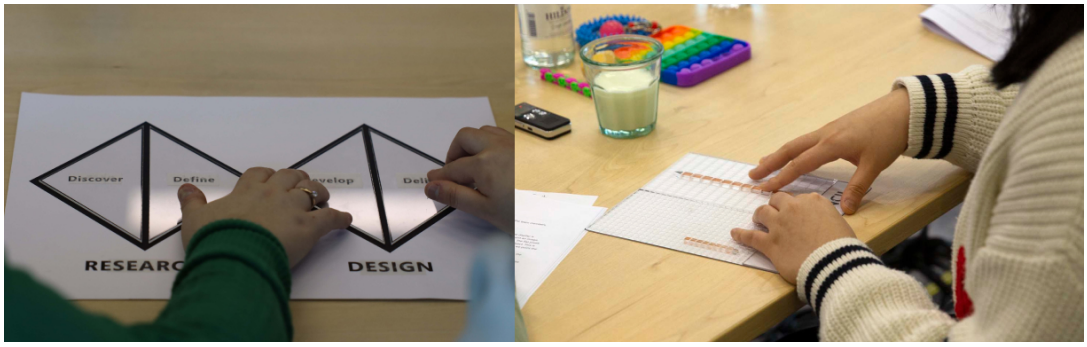


**Figure 2: (left) Tactile depiction of the double-diamond design model; (right) tactile phone screen to teach computer vision concepts, such as occlusion and perspective.**

creating a citizen design team was to shift from positioning people from the blind community as users and testers, to positioning them as citizen designers and co-creators of a technology that they will ultimately use. This goal echoes the ethos of participatory design [3, 15] with a further focus on skill building. Over a four-month period, we hosted three day-long, in-person workshops with our eight citizen designers with equal attention to what the co-designers were taught about the design process and how that understanding could be useful for the design process for Find My Things (see Figure 2). The three sessions focused on: user scenario development, teaching experience, and finding experience.

Learnings were synthesized from the sessions in a range of ways. All activities done by the citizen designers were recorded and analysed, such as the think-aloud elements of building their prototypes. This analysis, for example, led to UI suggestions such as: "There should be vibration feedback because I may not want to have my volume up in public. I don't want to attract attention to myself" (P1). Recordings of prototyping activities were also reviewed for the embodied experience of the space and the relationship citizen designers had with their phone. We observed that the citizen designers who were braillists tended to hold the phone horizontally, while print readers were likely to hold the phone at a 45-degree angle. Prototypes and artefacts produced by the citizen designers, such as the 'scenarios of use,' were reviewed. Telemetry data was also collected and used to improve the performance of early prototypes.

## 4 FIND MY THINGS

Find My Things is a teachable object localisation experience that supports a person who is blind or low vision find their personal things in 3D space using a phone. Rather than working only for generic objects, Find My Things gives users the power to personalise the system to any object, including small objects such as keys, medium-sized objects like backpacks, as well as shape-changing ones like a folding guide cane. Find My Things has two parts of the experience – teaching and finding. Teaching is done to add a new 'thing' or object to the experience, while finding can be used to locate any of the taught objects. The teaching process guides the user to record four short videos of a target object. These serve as training data for a few-shot object recognition model which can be personalized on-device in a couple of seconds. The find experience allows a user to select an object and scan their phone around the environment until the app localizes the object. The app then provides audio, visual, and haptic cues to guide the user to within arm's reach of their object.

### 4.1 Scenario of Use

Dayla knows that she is constantly looking for her lip balm - sometimes she misplaces it and sometimes it rolls away. She starts the teaching process. She is asked to put her lip balm on a clean surface and bring her phone close to the lip balm and tap the screen. She slowly draws the phone backwards, hearing an auditory progress bar and then a completion sound. She is then asked to show another side of her object and repeat the process. However, her lip balm goes out of camera frame, and she gets vibration feedback and the phone says 'move left'. She moves until the feedback goes away, knowing that the app is making sure that it can see her lip balm. She is asked to take two more videos with the object on a chair, and on the floor. She doesn't even have to move away from the table. The whole process takes just a few minutes.

The next day, Dayla is leaving early in the morning to go to work. She packs her bag but can't find her lip balm. She opens Find My Things and taps "lip balm." She scans her phone over the side table but doesn't hear anything. Dayla thinks where else she might have left her lip balm. Knowing the app only sees objects in the near vicinity (4 meters), she then walks to the kitchen and scans the large dining table. She hears a beep that tells her the lip balm has been spotted. As she moves toward it, she hears beeping that progressively gets faster and higher in pitch to guide her towards her lip balm. She manoeuvres around the table, orienting to the pings as the lip balm goes in and out of frame. The vibration increases, the pitch increases, and soon she hears the success sound. She reaches for the lip balm which is just under the phone. She pops it in her bag and heads out the door.

This is one of four "hero" scenarios that we optimized for. The other three are: 1) finding keys that fall out of a pocket when reaching into the pocket to answer a mobile phone that is in the same pocket; 2) finding a backpack that a colleague has moved; and 3) finding an ear bud that has rolled off the table during a lecture.

## 4.2 Technical Description

*4.2.1 System Architecture.* There are four main parts to the Find My Things system. The **client app** is a standalone C# iOS app that allows a user to teach/update or find personal objects or read the tutorial. The **teaching pipeline** supports the collection and selection of images that are processed with an on-device personalisation algorithm to return a mean feature embedding for the object. The **object recognition model** is an on-device model consisting of a meta-trained feature extractor and a set of embeddings that are outputted by the personalisation algorithm – one for each object the user has added. The **localisation pipeline** is an on-device process that compares incoming camera frames with an object's embedding to identify hotspots. If the confidence level of a hotspot is above a certain threshold, then the 3D guidance process is initiated using calculations based on surface detection.

*4.2.2 Teaching Pipeline.* Users are asked to follow specific directions to take four videos with varied backgrounds and perspectives. A spatial anchor is placed on the object using ARKit when the user touches the object with their phone. This anchor is used to provide feedback to the user if the object moves out of the camera frame. It also helps in the selection of frames that are used to create the personalized model embedding. Users are asked to draw the phone away from the object towards their shoulder until the requisite number of frames has been reached. Frames are sampled each time the camera moves 2mm, until 200 frames (per video) have been collected; this ensures that good variation in distance and perspective is gained. While users cannot replace specific videos, they can easily re-teach an object in just a few minutes.

The personalisation algorithm is launched and runs in the background each time a user finishes teaching a new object. The selected subset of 80 (20 per video) frames is fed through the object recognition model's feature extractor, and the resulting embeddings are averaged to obtain a mean embedding for that object. It takes on average 3 seconds on an iPhone 12 Pro, and 8 seconds on an iPhone 8.

*4.2.3 Object Recognition Model.* Find My Things is based on a few-shot image classification approach called Prototypical Networks [16]. The model consists of 1) a meta-trained feature extractor, and 2) a set of object prototypes (i.e., class-wise mean feature embeddings) – one for each of the user's objects. Together, they form a user's 'personalised' object recognition model and are stored as a single CoreML file on the user's device. The feature extractor is an EfficientNetB0 with 4 million parameters that has been trained on the ORBIT dataset [11] using an episodic training regime [4]. The resulting feature extractor can produce strong, linearly separable embeddings for a given set of objects using frames from only a few teaching videos per object.

*4.2.4 Localisation Pipeline.* We developed a localisation algorithm which would be more light-weight, and hence faster, than a traditional object detection model. Specifically, we perform a tree search on a particular frame, taking crop boxes of different sizes that can be passed through the user's personalised object recognition model. Each box has a confidence value, and if the value is above a (medium) threshold, the box is used to determine the likely location of the object in the frame. We average the centre pixel coordinate of each of these likely boxes, weighting by their confidence values. This gives us an estimated coordinate for the centre of the target object in the frame. In the case where this coordinate falls in a box with a confidence value of a second, higher, threshold, we use either LiDAR or ARKit's surface detection to convert the coordinate into a 3D location and initiate the guidance to direct the user towards that location.

This approach, as shown in Figure 3, can locate an object to a high degree of accuracy up to 4 metres away with an inference time of 100-200ms per frame. A start over button is also provided for the user to clear the current medium- and high-chance locations in cases that they suspect they're being guided in the wrong direction.

## 4.3 Key Learnings

*4.3.1 Understand the quantity and quality of examples required for optimal AI system performance.* Through experiments, we found that teaching examples that contained real-world quality issues, such as camera motion blur and the object being partially out-of-frame, lead to more robust model personalisation compared to teaching frames with no quality issues. We surmise that this is because there are quality issues during usage, hence the training data distribution matches the test data distribution more closely. Additionally, we found that more training examples did NOT lead to more reliable recognition of an object. We hypothesize that because teaching examples often contained quality issues, more teaching frames may reduce the signal-to-noise ratio, leading to a 'messier' representation of that object in the embedding space. Knowing the quantity and quality of examples required for optimal AI system performance informed our design of the teaching process. For example, we limited the number of teaching videos to four, as more videos would reduce system performance.

*4.3.2 Support users in providing examples in a structured way that reduces cognitive load and avoids over-guiding.* A teachable system brings flexibility, but also requires effort to teach. Therefore, as we designed the teaching process, we set it up in a way that is structured and prescriptive to reduce cognitive load and effort. For examples, we ask users to take videos of their objects rather than photos to reduce the (perceived) effort of blind users in taking "good" images. To help users keep the object in frame, we place an AR anchor on the object and only notify them when the anchor (as a proxy for the object) moves out of frame. Our citizen design team pointed out that constant guidance was cognitively demanding and stressful. Hence, this design approach ensures that users do not have to be concerned with something that they cannot necessarily judge - whether the object is in frame. Additionally, instead of using a freeform method which asks users to "show" us the object, we ask users to pull the camera back towards themselves, a body reference that all users could relate to. This simple drawing out method helps to collect multiple perspectives of the object from various distances. It also relieves users of the burden of having to think about how to best frame the object.

*4.3.3 Rapid teaching loop.* We aimed to reduce the time between teaching an object and testing it out, since users could leverage their knowledge of where the object is to test the app's performance. This gives users the opportunity to judge system performance for

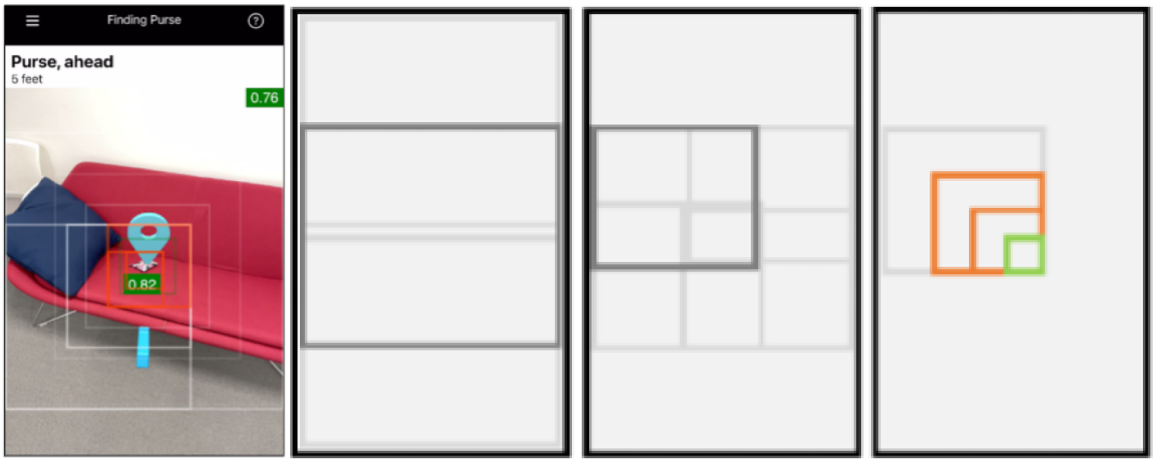


**Figure 3: Visualization of the localization algorithm used to find a purse. (left) visualization of crop boxes to localize the purse; (left middle): grey crop boxes of the tree search that continue to subdivide; (right middle) a focus on the crop boxes that have the highest confidence; (right) the orange boxes meet the medium confidence threshold and the green boxes the high confidence threshold used to trigger the find user experience.**

themselves by their own standards. Users can also use the environment to consider edge cases and thus better understand the boundaries of the system, something that users often forget [6]. To enable this, users could teach only one object at a time, with an experience flow that takes them straight back to the find screen so that they could test their object immediately. Additionally, we removed the need to calibrate the model after training to speed up the process.

## 5 DISCUSSION

AI has much to offer in enabling accessibility if experiences can be personalised to the needs of diverse users who have disabilities, addressing the long-tail distribution of user needs. Very recent advances in AI, such as foundation models, bring us even closer to meeting those diverse needs by increasing the number of tasks that a single model can do; however, the ways that we achieve the necessary personalisation of an experience have been given less attention. Teachable AI, for which users provide examples or high-level constraints to teach a model, has been proposed as a solution [7]. Yet, there is much generalizable design detail that can be learned from building and deploying a fully working end-to-end system.

In this interactivity, we present Find My Things, an application that allows people who are blind or low vision to find their personal items. To our knowledge, it is among the first fully realized end-to-end examples of a system applying Teachable AI to extend applications to the long-tail distribution of user accessibility needs. In the case of Find My Things, it extends object recognition to any personal item a user might own. One could imagine many more accessibility applications that could benefit from personalisation from text input /output to the way audio description/captions are provided in virtual reality and beyond. As the long distribution of user needs is a significant challenge in creating useful, scalable accessibility applications, we demonstrate how a teachable approach can address these challenges.

## REFERENCES

[1] Dragan Ahmetovic, Daisuke Sato, Uran Oh, Tatsuya Ishihara, Kris Kitani, and Chieko Asakawa. 2020. Recog: Supporting blind people in recognizing personal objects. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[2] Saleema Amershi, Maya Cakmak, W. Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI magazine* 35, 4: 105–120.

[3] C. Andrews. 2014. Accessible Participatory Design: Engaging and Including Visually Impaired Participants. In Inclusive Designing. Springer International Publishing.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 2017 ICML International Conference on Machine Learning*, 1126-1135.

[5] Jonggi Hong, Jaina Gandhi, Ernest Essuah Mensah, Farnaz Zamiri Zeraati, Ebrima Haddy Jarjue, Kyungjun Lee, and Hernisa Kacorri. 2022. Blind users accessing their training images in teachable object recognizers. *ASSETS. ACM Conference on Assistive Technologies* 2022. https://doi.org/10.1145/3517428.3544824

[6] Jonggi Hong, Kyungjun Lee, June Xu, and Hernisa Kacorri. 2020. Crowdsourcing the Perception of Machine Teaching. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.

[7] Hernisa Kacorri. 2017. Teachable machines for accessibility. *ACM SIGACCESS accessibility and computing*, 119: 10–18.

[8] Hernisa Kacorri, Kris M. Kitani, Jeffrey P. Bigham, and Chieko Asakawa. 2017. People with visual impairment training personal object recognizers: Feasibility and challenges. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 5839–5849.

[9] Simon Katan, Mick Grierson, and Rebecca Fiebrink. 2015. Using interactive machine learning to support interface development through workshops with disabled people. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/2702123.2702474

[10] Kyungjun Lee, Abhinav Shrivastava, and Hernisa Kacorri. 2020. Hand-priming in object localization for assistive egocentric vision. *IEEE Winter Conference on Applications of Computer Vision. IEEE Winter Conference on Applications of Computer Vision* 2020: 3411–3421.

[11] Daniela Massiceti, Luisa Zintgraf, John Bronskill, Lida Theodorou, Matthew Tobias Harris, Edward Cutrell, Cecily Morrison, Katja Hofmann, and Simone Stumpf. 2021. ORBIT: A real-world few-shot dataset for teachable object recognition. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/iccv48922.2021.01064

[12] Yuri Nakao and Yusuke Sugano. 2020. Use of Machine Learning by Non-Expert DHH People: Technological Understanding and Sound Perception. In *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society*. 1–12.

[13] Gonzalo Ramos, Christopher Meek, Patrice Simard, Jina Suh, and Soroush Ghorashi. 2020. Interactive machine teaching: a human-centered approach to building machine-learned models. *Human-computer interaction* 35, 5–6: 413–451.

[14] Téo Sanchez, Baptiste Caramiaux, Jules Françoise, Frédéric Bevilacqua, and Wendy E. Mackay. 2021. How do People Train a Machine? *Proceedings of the ACM on human-computer interaction* 5, CSCW1: 1–26.

[15] Dan Shapiro. 2005. Participatory design: the will to succeed. In *Proceedings of the CC Conference on Critical Computing*. 29–38.

[16] Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems* 30, 1 - 11.

[17] M. Steven, Ping Goodman, Dhruv Liu, Emma J. Jain, Jon E. Mcdonnell, and Leah Froehlich. 2021. Toward User-Driven Sound Recognizer Personalization with People Who Are d/Deaf or Hard of Hearing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5: 1–23.

[18] Mingxing Tan and Quoc V. Le. 2019. EfficientNet: Rethinking model scaling for convolutional Neural Networks. *arXiv [cs.LG]*. Retrieved from http://arxiv.org/abs/1905.11946

[19] Luisa M. Zintgraf, Kyriacos Shiarlis, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. 2018. Fast Context Adaptation via Meta-Learning. In *Proceedings of the ICML Conference on Machine Learning, 7693-7702.*