Understanding Personalized Accessibility through Teachable AI: Designing and Evaluating Find My Things for People who are Blind or Low Vision

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

Linda Wen

Microsoft Research, t-wenlinda@microsoft.com

Edward Cutrell

Microsoft Research, cutrell@microsoft.com

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, as long as the effort of providing examples is balanced with the benefit of the personalization received. As an example, this paper presents the design and evaluation of Find My Things, an end-to-end application that can be taught by people who are blind or low vision to find their personal things. Through synthesis of the design process, this paper offers design considerations for the teaching loop that is so critical to realizing the power of teachable AI for accessibility.

CCS CONCEPTS •Human-centered computing~Accessibility~Accessibility systems and tools • Human-centered computing~Accessibility~Accessibility design and evaluation methods

Additional Keywords and Phrases: Accessibility, Artificial Intelligence, Teachable AI

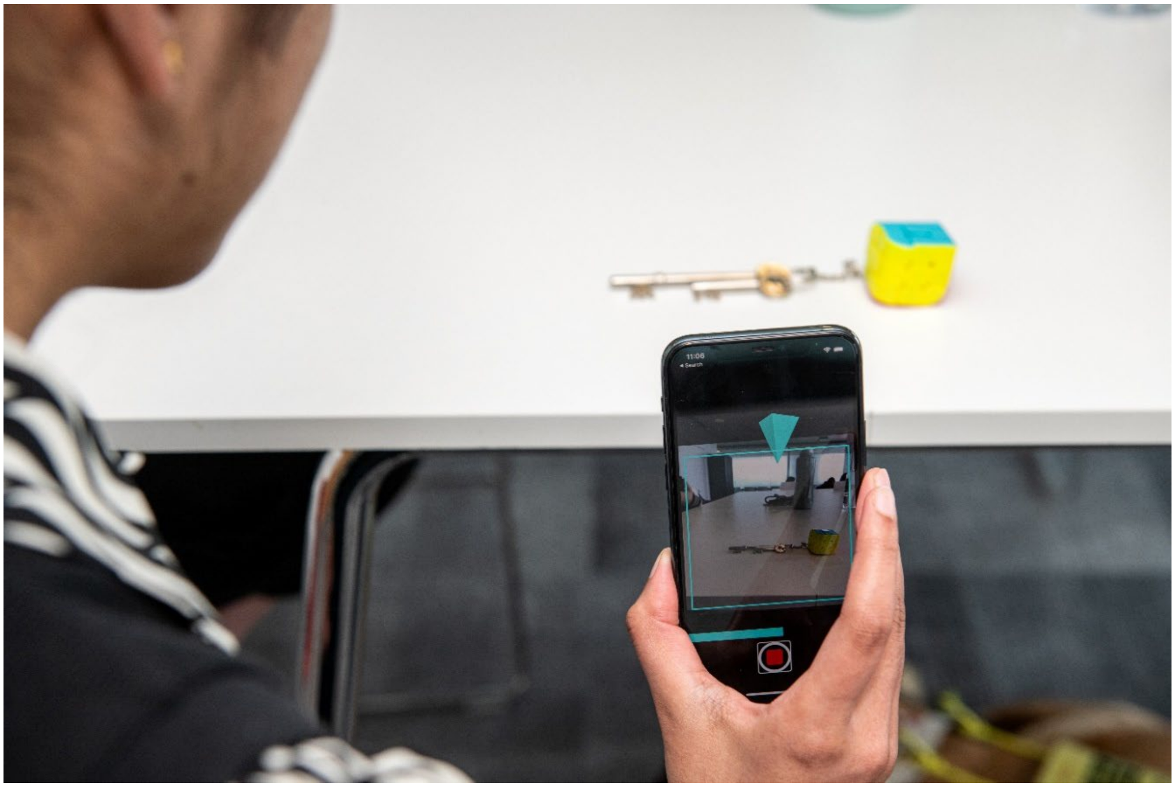

Figure 1: A user finds a set of keys with Find My Things, having previously taught the keys to the app through providing four videos.

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, e.g., GPT-4[1]. For example, the field of computer vision has already produced AI models that underpin a significant number of new visual access tools that have been adopted by the blind community. These range from talking cameras that read out short text through to the application of augmented reality capabilities to support indoor navigation, such as Seeing AI[2]. Yet, 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[3].

The diversity of user need in the disability community can present a challenge to creating broadly usable and efficacious AI systems for accessibility. There is large variation in user need both across and within disability categories. Those who are low-vision for example, may need a different user experience than those who are blind (as shown in [5,35]). Further,

[1] GPT-4 is a large multimodal model that accepts image and text inputs and emits text outputs that exhibits human-level performance on various professional and academic benchmarks. https://openai.com/research/gpt-4

[2] Seeing AI is a talking camera app that narrates the world around the user. https://www.microsoft.com/en-gb/ai/seeing-ai

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

those who are born blind may have differing capabilities and needs than people who become blind later in life. We can also think of differing personalities (e.g., [34]) and intersectional disabilities (e.g., [21]) that change user needs. The result is a very long tail distribution of user needs that must be accounted for in the design and development of AI systems for accessibility.

Teachable AI systems give users with disabilities a way to leverage the power of AI to personalize applications for their specific needs [14]. 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., [25]). 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. The teaching loop is a critical element to successfully realizing the opportunity that Teachable AI affords to accessibility.

We ground our study of teaching loops in the user-centred design and evaluation of *Find My Things*, an application to help people who are blind or low vision locate their personal items. As shown in Figure 1, a user is 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 allows scaling beyond the relatively small number of objects found in large image datasets, (e.g., 1000 in ImageNet [26]) to meet individual needs, from finding long guide canes to toothpaste caps.

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 needs of a more diverse set of users. It also allows us to consider in detail the design of the teaching loop, which requires an important trade-off between the effort of teaching (e.g., understanding what constitutes a good example) and the benefits of personalization in the experience [2]. Building on the findings of previous work [1,15], we present an evaluated end-to-end solution. We further deepen the learning through a description of the ways new machine learning approaches, experience design, and the voices of a citizen design team came together in the iterative development of the teaching loop. In brief, our contributions are:

1. Find My Things, an end-to-end teachable object recognition app that can be taught by people who are blind or low vision to find their personal things;
2. Detailed description of the creation and evaluation of Find My Things that captures the design and machine learning choices made in conjunction with the citizen design team.
3. Design principles for developing teaching loops for Teachable AI applications for accessibility synthesized from the learnings of the development process.

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 [2,25]. The rapid, incremental interaction cycles encourage a close coupling between user and resultant machine learning model. Applications are as far-ranging as optimizing web search [3] to creating classifiers to detect melanoma cancers [10] or creating novel musical instruments [17].

One of the key challenges of interactive machine learning systems is supporting the mental model of the user during the interactive process of refinement. In [8], the authors observed that users, when building gesture-based musical instruments, employ typical machine learning evaluation techniques, such as cross-validation, both to improve their machine learning model as well as to learn how to provide better training data. End-user programming research has offered the concepts of selection and coordination barriers that characterize challenges in how users decide 'what' element in the data to change and 'how' to change it [18]. Both seminal works speak to what we would call a *teaching loop*, the back and forth between user and ML model to get to the desired end result.

Several papers have explored teaching loops in detail. In the context of machine-taught perception (like teachable object recognizers [13]) authors found that most participants intuitively understood that they should provide a variety of examples that captured discriminative parts, e.g., the logo on the product. About half of participants understood that the types of variation and quantity of examples should be consistent across classes (e.g., objects), but few focused their effort on the diversity of their test set (e.g., by including edge cases). These findings are confirmed in [27] that propose the following guidelines: provide guidance for building teaching sequences; allow modifications to past teaching actions and sequences of actions; assist the data augmentation process; and show optimization inertia and model state changes. These findings suggest a need to focus on more than just collecting good quality data to make the teaching loop work well.

Interactive machine learning, in its various manifestations, has situated the user in relationship to the technology in subtly different ways. Search and recommender systems both offer users ways to offer feedback through change of query or binary responses on specific questions. The user, however, may not perceive their role or agency in shaping these machine learning systems. In contrast, the release of Teachable Machine [6], which underpins a wide variety of experiences, more directly emphasizes the agency of the user in creating the final AI experience / system through their role as teacher and providing examples.

2.2 Teachable AI for Disability

Teachable AI for disability has been proposed [14] as a mechanism to give people with disabilities the agency to personalize experiences to their own needs and situations. It could be adapting previously inaccessible tools or making a new class of tools. Experiences explored in the research literature have enabled people with learning and physical disabilities to use electronic music interfaces [17], as well as a personalized sound recognizer for people who are deaf or hard of hearing [11,23]. However, most examples of teachable AI for disability have been teachable object recognizers for people who are blind or low vision, e.g., [1,12,15].

The earliest exploration of the potential of teachable object recognizers asked users to collect 50 images each of several objects at home to verify the need for teachable object recognizers [15]. Key to these findings were that users found the idea of distinguishing between items that felt similar, such as different bottles of shampoo, particularly compelling. The authors also coupled this data collection exercise with a lab study that asked participants to train and test several objects in a laboratory experiment. Initial data capture showed that participants needed guidance in taking their images, as many used extreme points of view. Classifier performance differed dramatically across participants but was best on the participants' own photos and too many photos decreased performance. Participants were also very concerned about the quality of their images.

Follow-up work has looked at different strategies to guide the taking of images. In ReCog [1], ARKit[4] along with the camera position or its motion is used to calculate the position of an object. Sonified and verbal feedback are used to direct the extent and direction of needed movement to take a good image. In contrast, [20] exploits hand-to-hand referencing used in non-visual engagement to localize an object based on semantic information from the segmentation of a hand placed

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

near or holding the object. Work has also addressed how people who are blind can access the content of their training images [12]. These methods provide a range of different starting points for designing the teaching loop.

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., [30]. This in turn enables AI models to more readily 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 [31]; 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., [36]. Models that are quick to adapt and have fast inference times are important to achieving interactive AI experiences.

Only recently has few-shot learning matured enough to be applied to real-world challenges. State-of-the-art performance on simplified datasets, such as the characters of Omniglot [19] or the high-quality images of miniImageNet [33], is now relatively saturated [7,24]. To drive further innovation in few-shot learning, the focus is shifting to real-world data, made possible by the collection of new datasets, such as ORBIT [22]. 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. With its associated benchmark, it provides a rich playground to drive research in robust few-shot learning. The advances in few-shot learning and the publication of the ORBIT dataset provided the foundation for developing the Find My Things app.

3 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, shown in Video Figure 1. Rather than working only for generic objects, it 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.

3.1 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.

3.2 Technical Description

3.2.1 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.

3.2.2 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.

3.2.3 Object Recognition Model

Find My Things is based on a few-shot image classification approach called Prototypical Networks [30]. 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 [22] using an episodic training regime [9]. 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.

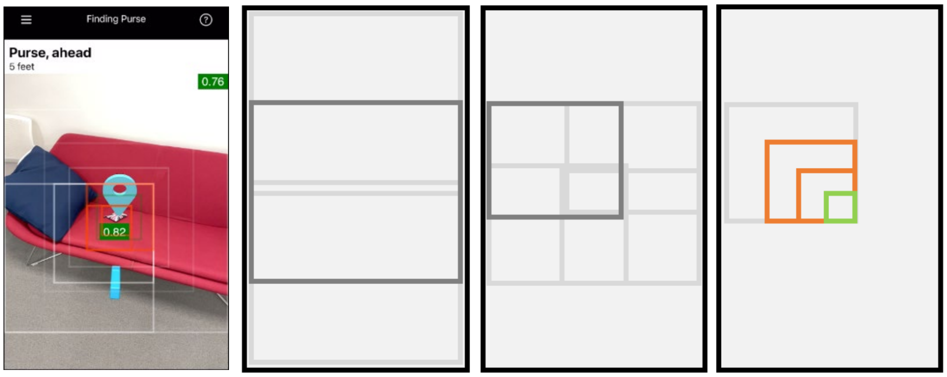

Figure 2: Visualization of the localisation algorithm used to find a purse. (left) visualization of crop boxes to localize the purse; (left middle): grey crop boxes of the tree search that continue to subdivide; (right middle) a focus on the crop boxes that have the highest confidence; (right) the orange boxes meet the medium confidence threshold and the green boxes the high confidence threshold used to trigger the find user experience.

This final model was selected through a three-phase process. In the first phase, we ran ~790 independent experiments on the ORBIT dataset to identify the best set of hyperparameters, including the choice of feature extractor, video and frame sampling methods, loss function and optimizer. This yielded a set of 10 AI model candidates which had the highest average frame accuracy on the ORBIT test set, as well as strong performance on metrics that were meaningful to the user experience. These included the time (in seconds) to personalise the model to a user's new objects, and the inference time through the personalised model.

In the second phase, we compared the 10 candidate AI models on a dataset collected directly with a modified version of the Find My Things app. We recorded the object's initial position in 3D using ARKit and then "found" objects per our hero scenarios. Knowing when the object was in frame and its distance to the camera, a wider suite of metrics was then used in a scorecard approach. Effort was made to maximize true positives (when an object was in view and recognized) as well as minimize false positives (when an object was recognized but not in view). These numbers were considered both at medium and high confidence thresholds which respectively triggered the localization of a hot spot and guidance parts of the experience.

In the third phase, the final 5 candidate AI models were ported into the Find My Things app and compared in live side-by-side tests: research team members held up 2 phones, each with a different model in the backend, and searched for the same object in one of the hero scenarios that directed the design. The testers conducted pairwise comparisons for each candidate in 3 different scenarios, repeating this twice. They found small differences in inference time that made the user experience sluggish or frustrating; they also noted that some AI models were able to recognize objects better at a distance. The AI model with the best performance in overall user experience as judged by the team was selected as the final model, as numerical comparisons were judged as not meaningful to the overall experience.

3.2.4 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 2, 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.

3.3 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. They applied to participate through the VICTA charity in the UK, which runs events to support the learning and confidence of young people who are blind or low vision. Our cohort consisted of three braillists and five print users, using screen reader technology and magnification respectively, to access their phones. All were young people who confidently (and continuously) used their phones. The braillists had all used Seeing AI previously, but those who were print users had less experience with vision-specific assistive technology.

Inspired by the concept of citizen science [29], we wanted to engage young people with a range of vision levels to learn about the design process through apprenticeship to a professional technology development team. As design is often taught visually, it can be unavailable to young people from the blind community, reducing the number of technology designers with lived experience of blindness. Having selected participants who already showed an interest in design and engineering, it was our hope that this experience might allow them to grow their abilities and later pursue careers as technologists.

Key to our perspective in developing a citizen design team was shifting from seeing people from the blind community as our users and testers, to seeing our citizen designers as co-creators of a technology that they will ultimately use, similar to participatory design [4,28], with a further focus on building skills. 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 used to further the design process for Find My Things. The three sessions focused on: user scenario development, teaching experience, and finding experience. The citizen designers also participated in formative and summative user studies to observe the whole design process.

Each session had a similar format. In the mornings, a spark activity was used to get people thinking about a design dilemma, followed by an educational session. We covered: the double diamond design model, personas and scenarios, rapid prototyping, usability testing and A/B testing. We also ran sessions on how computer vision systems work. The afternoons were devoted to a range of prototyping sessions, in which our citizen designers could try out different combinations of design elements in a live prototype to find, and reflect on, an experience that worked well given the constraints of the technology. For many of the sessions, we made tactile resources, such as depictions of the double-diamond design model or tactile phone screens as shown in Figure 3. The full protocol for each citizen design workshop can be found in the supplementary materials.

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

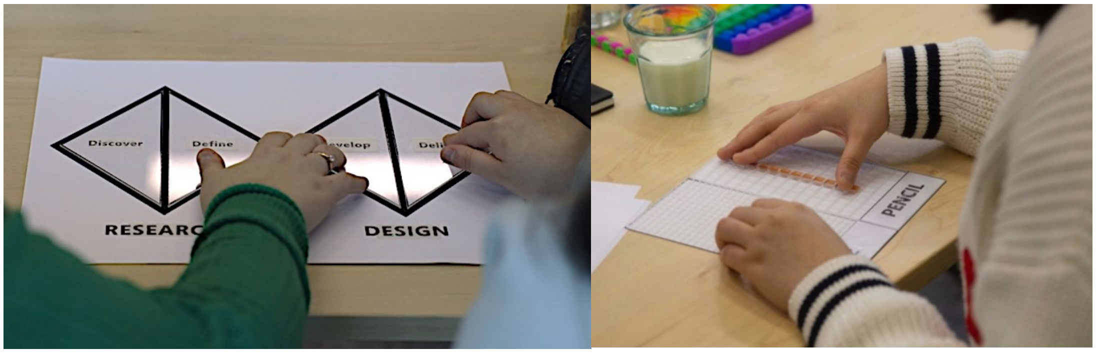

Figure 3. (left) Tactile depiction of the double-diamond design model; (right) tactile phone screen to teach computer vision concepts, such as occlusion and perspective.

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 and telemetry data was also collected and used to improve the performance of early prototypes.

Beyond the delivery and accessibility of learning content, the agency that citizen designers had in the process was an important contributor to the success of the engagement. Citizen designers could truly feel part of the team and influence the development of the technology in real-time. The research team worked iteratively (over the four months and three workshop sessions) to build the experience from the ground up based on the learnings from each session. In each session the participants were able to see evolution and how their contributions informed the AI system and the overall experience. The findings of these sessions are incorporated into our discussion of the teaching loop, presented in the next section.

4 DESIGNING THE TEACHING LOOP

In a teachable AI system, the aim of the teaching loop is to support users in providing examples to an AI system for the purpose of helping them reach a personally desired system outcome. We argue that this is not a matter of just providing "good" data, but helping the user build a mental model of what "good" might be in their own context. This empowers the user to adjust system performance for their own needs. We might think of the teaching loop as a literal loop in which a user provides examples, tests the AI system, and then adds or changes the provided examples. However, our design process illustrated that the teaching loop, or the iterative engagement between user and ML system, can take many forms. We reflect upon some of the ways that happened in the design of Find My Things.

4.1 Realistic Examples

We began our design explorations of the teaching loop by asking the question: what kind of teaching examples lead to the most effective personalisation, and hence the best performance in recognising the user's objects in test scenarios? 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 personalization compared to teaching frames with no quality issues. We surmise that that this is because there are quality issues during usage, and so the training data distribution more closely matches the test data distribution. This result aligns with findings from [11,12,15] which demonstrate the importance of consistency between data captured for teaching and the data that will be captured as part of the experience itself.

We conducted analyses which controlled for the proportion of a user's teaching frames that were marked with one or more quality issues, such as blur or framing issues. We then compared the average frame accuracy after model personalization using different proportions across each of these different settings. We found that, overall, the model had the highest average accuracy and lowest variance (i.e., was most robust) when the teaching frames contained both framing *and* blur issues, compared to only frames with one or the other or no issue. In particular, robustness peaked when 60-80% of teaching frames had both quality issues present. The model performed least well, with lowest average accuracy and highest model variance, when trained only on 'perfect' teaching examples as shown in Table 1.

Table 1: Some real-world noise in the teaching examples is important for robust personalization: personalizing with teaching examples that have quality issues (e.g., blur or poor framing) results in higher test accuracy and lower variance, compared to personalizing with teaching frames that are all perfect. Results are reported for 158 objects from the ORBIT test set, where each run is repeated 50 times per object.

Proportion of teaching frames with a blur and framing issue	0% (all perfect)	20%	40%	60%	80%	100% (all imperfect)
Average test accuracy (variance) %	90.5 (24.5)	96.0 (16.0)	96.5 (13.1)	96.9 (12.1)	97.0 (12.2)	96.8 (13.3)

However, we also found that more training examples did NOT lead to more reliable recognition of an object. We conducted analyses which controlled for the number of teaching frames (and number of videos) per object and found that 10-20 frames per video led to peak performance. Sampling more than 20 frames per video, in fact, reduced performance by 3-4 percentage points. Our analyses revealed that this was occurring because a user's mean feature embeddings became less linearly separable with increasing numbers of teaching examples. Specifically, the inter-class variance between the user's mean feature embeddings reduced and the intra-class variance between the features used to compute the embedding increased. We hypothesized that because teaching examples often contained quality issues, more teaching frames might be reducing the signal-to-noise ratio, leading to a 'messier' representation of that object in the embedding space. As such, we can limit the amount of data needed in the teaching loop.

4.2 Dynamic Support

We took a human-centered approach to considering how we supported users in providing example images for teaching their object. We started by asking and role-playing with our citizen designers what strategies, skills and needs users might have when using the camera to record an object. It quickly became clear that video was much more flexible than still images. It allowed the system to pick the best images for training, rather than insisting that the user take "good" images. The teaching process then developed to build on existing embodied strategies common to blind users – using their hands and body to orient the camera. First, the user touches the object with the phone camera and then draws back towards their shoulder.

We decided to build an experience that would ask the technology to help the user if the object went out of frame, as this was the one quality issue that had a detrimental effect on the AI system. First, we place an AR anchor on the object when the user touches their phone to the object and hits start. As they draw their phone back from the object, we provide feedback if the anchor (as a proxy for the object) moves out of frame. This stands in contrast to approaches in which guidance is provided to keep the object "in" frame (e.g., [1]). Our citizen design team pointed out that a bit of feedback can help the user feel good about their efforts, while constant guidance was cognitively demanding and therefore stressful. 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.

We chose to ask users to pull the camera back towards themselves, using a body reference that all users could relate to. We decided not to ask users to use a "free form" method to "show" us the object. When comparing approaches, we found that the citizen designers had to think a lot harder about what in the object they needed to show when it was freeform. It also made automatic selection of diverse images much harder due to variation between users. We also found that free-from example collection did not improve model outcomes and was slower for the user. The drawing out method helps users get multiple perspectives in the examples (and distances) without asking users to imagine what these could be and figure out how to frame and take them. Other research has also shown that users often try extreme perspectives which is detrimental to system performance [15].

To further increase the variety of images, we walk the user through repeating the example capture technique four times with different backgrounds and object rotations. While initially we did not specify where the backgrounds could be, we found that thinking of these possibilities was mentally taxing and required users to move around a space. In response, we shifted the instructions to focus on table, chair, and floor that could be right next to them, speeding up the process and decreasing cognitive load. We also added object rotation as only some of our citizen designers came to understand how necessary this might be. We limited the number of examples to four as several users attempted to improve recognition by increasing the number of examples, which actually reduces performance.

Some citizen designers did make a direct association between how they wanted to find the object and how they should take the videos. For example, one citizen designer provided videos of her headphones at a distance in hope that the app would recognize them better at a distance. While this is ideal user behaviour, unfortunately, in this case it is not a correct mental model. Through work with our citizen design team, we aimed to structure the experience to encourage a correct mental model through both the design of the recording technique as well through the creation of written materials (e.g., app descriptions and troubleshooting tips). For example, we encourage teaching visually distinct categories (e.g., keys and wallet), rather than using Find My Things to distinguish between similar things (e.g., a red and green marker) as early AI model testing showed that the AI model produced wrong predictions with very high confidence in distinguishing tasks.

4.3 A Rapid Loop

We aimed to reduce the time between teaching an object and testing it out. For example, we designed the teaching process so that users could teach only one object at a time, with an experience flow that took the user straight back to the find screen so that they could test their object immediately. During the citizen design sessions, we observed that citizen designers were able to relate their teaching videos to how well the system was able to recognize the object. One citizen designer noted that simple objects (e.g., a ball) taught on the same colour background (e.g., a white table) were not well recognized by the system, but more complex ones were (e.g., keys). By designing a tight loop, a user is able to connect how their actions during teaching may influence the result.

The ability to test immediately after teaching worked well as the user could leverage their knowledge of where the object was in order to test it. This gives users the opportunity to judge good or bad recognition *for themselves*, using their own standards and allowing for exploration of performance. For example, instead of an overall performance metric, they are able to test how well the model performs on their carpet versus on their coffee table; from far or from up close; moving the phone quickly or slowly. This process allows the user to go beyond a numerical understanding of how well the system is performing and get an insight to when and how the system might perform well or poorly and how they can optimize it for their own needs. It was for this reason that we enforce teaching only one object at a time.

The type of feedback that users receive after teaching was also considered. Importantly, we needed to communicate to the user how well the model that they created would work in practice. Our citizen designers wanted a sense of how fast their object would be recognized when in frame, from how far away, or whether it would be recognized against their tartan carpet, for example. A simple metric such as accuracy was therefore not appropriate. We also explored telling them whether a certain example was impacting the performance of the model. However, we found that our citizen designers much preferred the interactive and insightful experience of just using the app immediately after teaching something.

To achieve a rapid teaching loop, we needed to focus on decreasing the time it took the AI model to personalize. We see in Table 2 that personalising the model by fine-tuning it on the user's teaching videos has a significantly longer personalization time than when using a ProtoNets approach. This is because fine-tuning the model involves 50 backward-forward passes (i.e., gradient steps) through the model, while ProtoNets involves only a single forward pass. Most importantly, we discovered that some larger models have slightly higher accuracy on the ORBIT test set but were slow to personalize. While the VIT-B-32 (CLIP) and EffecientNetB0 feature extractors have similar personalization times on a 32GB NVIDIA V100 GPU, ViT-B-32 (CLIP) was dramatically slower on a phone. Rapid personalization enables users to throw out a model that isn't working well and try again, which requires much less user effort than trying to understand uncertainty measures provided by the model.

Table 2: Larger feature extractors (EfficientNetV2, ViT-B-32) were not able to support a rapid teaching loop on a mobile phone as well as an EfficientNetB0: although the larger extractors had similar personalisation and inference times to an EfficientNetB0 on a GPU, when ported onto a mobile phone, these times increased dramatically. Personalisation times were also an order of magnitude larger when personalising the model with a fine-tuning compared to a ProtoNets-based approach. Times were measured on a 32GB NVIDIA V100 GPU.

Feature extractor	Pre-training dataset and method	No. of parameters	Personalise method	Avg. frame acc (%)	Time to personalise on GPU (s)	Inference time on GPU (µs)
EfficientNetB0	ImageNet1K (supervised)	4.01M	ProtoNets	69.8	2	186
			Finetuning	68.8	36	186
EfficientNetV2	ImageNet21K (supervised)	20.18M	ProtoNets	73.8	3	835
			Finetuning	71.6	62	835
ViT-B-32 (CLIP)	LAION2B (CLIP; self-supervised)	87.46M	ProtoNets	74.8	2	193
			Finetuning	73.8	55	193

5 USER EVALUATION

A user study was designed to provide structured data collection from 'in the wild' contexts, generating qualitative, quantitative, and log data of these experiences. We triangulated this data to understand whether the cumulative design and machine learning decisions made led to an appropriate balance between the effort of teaching and the benefits of personalisation shown through successful usage. We ask the following three research questions:

R1: Are people able to successfully use Find My Things?
R2: What are the potential benefits of this teachable system, Find My Things?
R3: What kind of effort does it require to make Find My Things work?

5.1 Study Design

5.1.1 Participants

The user study included 15 participants who are blind or low vision. The majority were opportunistically recruited from the back-to-work program run by the Canadian National Institute for the Blind (CNIB). These participants ranged in age from 18 – 65, with representation from all age groups. Half of the participants used Apple VoiceOver technology, while the others used magnification or a combination of both. All were regular smartphone users and had previously used Seeing AI. About one-third regularly tested technologies for CNIB and were particularly proficient. This range of familiarity is what we might expect in users of assistive AI systems and therefore suitable to helping us understand usage. Gender was evenly split. An additional three participants (2 female) were recruited from our citizen design team, ranging in age from 14 – 25. These included a VoiceOver user, a magnification user, and a deaf-blind participant who used magnification. They were all Seeing AI users and very proficient at using their smartphone for access.

5.1.2 User Study

The user study ran over a period of three weeks, in which participants: 1) attended a 30-minute briefing call; 2) completed a 1-hr mission 'in the wild'; and 3) used the app freely for one week. The briefing call introduced the Find My Things app and its purpose as well as clarified the study instructions that were provided in document format. The 1-hr mission was a specified task done by participants 'in the wild', that is, in their own space at their own time. Participants were asked to teach and find three objects, each inspired by one of our hero scenarios listed above. They were asked to find each object twice, encouraged in the second attempt to adjust the difficulty of the scenario depending on the success (or not) of the first try. Users were then asked to teach two more objects and use the app as and when it was useful in the following week. See supporting materials for full instruction set to users that communicated the protocol.

5.1.3 Data Collection and Analysis

We designed a mixed-methods study that could be completed fully 'in the wild' without the support of a researcher. As part of an in-usage survey, participants were asked three (obligatory) multiple choice questions immediately after each experience of finding an object. Participants were then invited to share more details about their multiple choice answers through open questions about the context of use (optional). In particular, through the mission protocol, participants were invited to share how they made their second 'find' attempt for each object easier or harder, revealing their mental model of how Find My Things worked. After all find experiences were complete, the study closed with a final survey. It included 5 Likert scale questions and 3 open-ended questions. See Table 3 for an overview of the questions.

The multiple-choice questions in the in-usage survey were tabulated, and the success rate for all participants were calculated based on the answers. If the user answered "Yes" to the question "did you find your thing?" and "Yes" or "Somewhat" to the question "did the app assist you in finding your thing?", then that specific attempt is counted as a success, meaning that the app was useful in assisting the user in finding their thing. Answers to the optional open-ended questions were used to contexualise the analysis of the recordings of each attempt. All results were considered in light of the open-ended questions in the final survey. Carrying out a thematic analysis, responses were coded as benefits, effort, improvements, and other. Improvements were then sub-divided into the relevant component (e.g. UI, localisation pipeline ect).

A range of log data was collected in addition to the survey data. This included timestamps and durations of the training and find experiences, possible object locations during a find experience and data tracking the orientation and movement of the phone during training and finding. The data allowed us to observe the duration of both teaching and finding and measure how users responded to the audio guidance. After creating a ground truth for the object location in the find videos, we were also able to determine how well the localisation algorithm was performing across a range of devices.

Table 3: Survey questions.

In-usage survey questions (after the completion of every find experience)	
Questions	Answer format
1. Did you find your thing?	Multiple choice between yes, somewhat, and no (obligatory)
2. Did the app assist you in finding your thing?	
3. Could you follow the sounds and/or visual feedback to find your things?	
1. Why were you looking for your thing?	Open-ended (optional)
2. Where were you looking?	
3. What surface was your thing on?	
4. What worked well?	
5. What was challenging?	

Final survey questions (when the participant finishes the study)		
Questions		Answer format
Please rate the following statements:	1. I was able to quickly learn how to teach new things to the Find My Things app	Multiple choice between strongly agree, agree, neutral, disagree and strongly disagree
	2. I feel more confident finding my things when I use the Find My Things app	
	3. I found the sound and/or visual guidance to my things difficult to follow	
	4. It is not important to me to be able to teach the Find My Things app about things that are important to me	
	5. I believe that the Find My Things app would help me be discreet when looking for things in public environments or work situations	
Please respond in your own words to the following questions	6. What is the main benefit of the Find My Things app for you?	Open-ended
	7. How could Find My Things be improved to meet your needs	
	8. How would you describe the Find My Things app to a friend?	

5.2 Findings

Overall, users succeeded in finding their personal objects and found Find My Things helpful. The app was used 116 times, 86 of which were regarded as valid runs in which no technical error occurred. Technical errors included the AR session not initializing correctly or the camera being occluded for long durations during find experience. In 71 of these 86 valid runs, participants stated that the app helped (63 of 71) or somewhat helped (8 of 71) them locate their objects, attaining a success rate of 83%. Based on the final survey results, 12 out of 15 participants reported feeling more confident finding their objects with Find My Things.

Beyond the user experience, we ran extensive analyses on the recordings from each search to quantify the performance of the object recognition model. We found that when the object was in-view and within 4 meters of the camera it correctly localized the object 72.4% of the time. Failure cases included: 1) things that were similar in shape and colour from far distances (e.g., a white piece of fluff rather than white MacBook charger), which often resolved as the user got closer to the thing; 2) objects in low-contrast scenarios – for example, if the surface was poorly lit, had a glare, or was the same colour as the object (e.g., a white AirPod on a white tile). In 13 valid runs, phones without LiDAR did not guide the user to the correct location.

Participants suggested several new use cases. These included: 1) finding more than one thing at a time (e.g., finding a set of objects needed for school); 2) finding a category of similar things based on only one taught item (e.g., teaching one dish towel to find other similar dish towels); and 3) finding an item that is not taught because the user did not expect to lose it. We also noted that the design could be improved to let users know that a technical error had occurred.

5.2.1 Benefits of a Teachable System

Users added a large variety of personal objects that differed greatly in appearance and functionality. A total of 58 objects, which could be grouped into 37 visually distinct categories, were added by 15 people. Twenty-five of these categories contained only one object, indicating a long tail of possible objects. Common objects include different kinds of keys (e.g., house, mailbox—7 occurrences) and earphones (e.g., Air Pods—3 occurrences). Less common objects are exemplified by guide dogs (2 occurrences), pliers (1 occurrence), dryer balls (1 occurrence) and braille stylus (1 occurrence). The varied functionality of these objects demonstrates the app's ability to assist users in a much wider range of scenarios than standard object recognition.

The value of being able to stay in charge of essential personal objects for our users was highlighted consistently through qualitative feedback. As one participant articulated:

> *I can find personal items using the app if the item is lost. I have about 5-10 personal items that I always have with me and most of them are essential. If I lose a bus pass, I can't get to anywhere, if I lose keys, I can't get home, etc.* - P1

While many users have strategies for finding items, often through diligent organization of items or tactile searching methods, qualitative feedback suggests that Find My Things can augment these strategies when in an unfamiliar place, bound by social norms, or short on time.

> *(The main benefit of Find My Things is) helping me find objects in public places where I'm not sure where my objects would be lost. Just open the app and it will scan for this object in the environment, it's like having new eyes!* -P15

Indeed, many users found that Find My Things helped them be more discreet when searching for things in public or formal situations. For these users, it helped them avoid touching the floor, which may not be clean or could be awkward in a work context. According to the final survey results, 9 out of 15 respondents agreed or strongly agreed with the statement: "I believe that the Find My Things app would help me be discreet when looking for things in public environments or work situations."

Finally, Find My Things can support users wanting more independence or having multiple disabilities. Seven out of 15 participants used words like "independence", "autonomy" and "not needing sighted assistance" when describing the benefits of Find My Things. Users also noted the ways it supported their strategies when managing multiple disabilities, such as deaf-blindness or additional memory issues. For example:

> *Since I have memory issues, I sometimes forget where I put things. It is very helpful to hear the sound, so that I know, approximately where I have to go to look, and I appreciate the guidance that is given.* - P8

> *If I drop something, I often cannot hear where it bounced off to due to my deafness, so the app would be helpful in that instance too.* -P1

This data suggests that the ability to personalize might particularly support those in the long tail of disability diversity.

5.2.2 The Effort of Teaching

A teachable system brings flexibility, but also requires effort to teach. Our data suggests that this level of effort was not prohibitive. All 15 respondents agreed or strongly agreed with the statement: "I was able to quickly learn how to teach new things to the Find My Things app," suggesting that it is easy to get a hang of the teaching process. On average, participants were able to teach an object in 2.4 minutes, though after 3-4 objects, this dropped to around 1 minute per object. Furthermore, only six examples did not fully adhere to the instructions to showcase different sides of the object and use various surfaces while recording training videos, which help to produce high-quality training videos for the AI model.

6 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 need. 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 personalization 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 [14]. Yet, this approach has not been validated in a fully working end-to-end system. As a result, there are few generalised learnings or design considerations that support the design of teachable AI experiences specifically for accessibility.

To address the lack of design considerations for Teachable AI systems for accessibility, we synthesised our learnings from the design and evaluation of Find My Things reported in this paper. To our knowledge, Find My Things, an application that allows people who are blind or low vision to find their personal items, 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. The evaluation shows that users can indeed find their personal things with the app in their own environments and that effort to teach the app is balanced with the benefit of being able to find personal (and not just generic) things.

6.1 Learnings

1. Understand the quantity and quality of examples required for optimal AI system performance

It is important to consider the quantity and quality of examples needed to produce the best performing AI system. It is easy to assume that "good" data can be equated to "lots" of "clean" data. In the image context, that would mean as many as possible well-lit images with the object of interest centred and no camera motion blur or obstruction. In contrast to expectation, but similar to the trend reported in Kacorri et al. [15], we found that small amounts of a user's own data led to the best performing AI system. This may be particularly true in accessibility applications in which the users' data may differ dramatically (i.e., be out of distribution) to the data that an AI system is trained on as shown in [22].

The findings in this paper underscore the importance of experimentally determining what constitutes "good" data for any given teachable AI system before beginning the design process. Indeed, what constituted "good" data in Find My Things versus Kacorri et al. [15] were not the same despite the application domain being the same. Indeed, small technical differences bring nuance to notions of "good" data. To ensure the external validity of these experiments, it is important to think about the metrics being used in these calculations. A user-centric approach to metrics might focus more on per-user performance rather than average accuracy across all frames in the test set. Variance over users and worst-case per-user performance are other alternatives that can bring a more human-centric framing to metrics being used to make user experience design decisions.

2. Support users in providing examples in a structured way that reduces cognitive load and avoids over-guiding

In many accessibility use cases, what constitutes a “good” example can be quite concrete. As a result, designers can reduce both the time and cognitive load of capturing good teaching examples through a structured experience. In the Find My Things example, video is used to reduce the (perceived) effort of users who are blind in taking “good” images. Further, the number of videos and the way they were taken were also defined to reduce the number of choices that a user had to make, and by consequence, the cognitive load. Users did not need to think about perspective or background changes, for example. They did not even have to move around in space, a potential challenge for some.

While structure is appreciated, the reporting of our design process showed that heavily guided experiences can also be cognitively taxing. Our citizen designers reminded us that it is better to know when you’ve gone wrong rather than constantly trying to follow feedback to do it right. Methods that build on existing skills, such as hand-to-hand referencing in the blind community, may also give users a stronger sense that their existing capabilities are being augmented rather than replaced [32]. The concrete design opportunity of teachable AI for accessibility stands in contrasts to teaching interactive machine learning for creative applications, the basis for most learnings and guidelines in the literature. In contrast to guidelines that encourage user experimentation, e.g., [18,27], we emphasize more experimentation by the AI system developers rather than placing this demand entirely on users in the accessibility context.

3. Shorten the feedback loop as much as possible and reduce friction to re-teach

A short feedback loop between a user teaching and trialing an AI system allows users to quickly test and iterate their personalized AI system. As users are often in a context of use when teaching, they are practically set up to trial the app. Users can also use the environment to consider edge cases and thus better understand the boundaries of the system, something that users often forget [13]. Shortening the feedback loop is dependent on the personalisation time of the model. It could also be about other components in the pipeline, for example, in our case, taking out the need to calibrate the model after training.

A short feedback loop also reduces the friction to re-teach a concept if something goes wrong. The ability to re-teach reduces system complexity by avoiding the need to provide the user with other signals of AI system interpretability, e.g., uncertainty values for videos. It is also important to consider other reasons that it might be difficult to re-teach a concept. Does the user need all their examples cached in order to retrain a model without catastrophic forgetting? If so, might there be privacy implications if these are stored in the cloud or practical limitations of phones if stored locally [16]? Might it be costly to retrain a model if the user is unsatisfied with the quality? The design process must engage with the potential AI system constraints when holistically trying to shorten the feedback loop and reduce friction for re-teaching.

6.2 Reflections

As AI becomes prevalent in accessibility experiences, taking a human-centred approach to the design of such technologies will be critical to creating experiences that really enable people with disabilities. We present an example of such an approach in this paper using a citizen design team. There is much opportunity for researchers to grow the methods of user-centred AI design for accessibility. Indeed, the methods developed in this inclusive space will likely shape the way research thinks about human-AI interaction, just as many technologies developed first for disabilities extend to enable all people.

7 CONCLUSION

Find My Things demonstrates the power of Teachable AI in a fully realized end-to-end system. In this case, we extend object recognition to any personal item a user might own. One could imagine many more accessibility applications that could benefit from personalization from text input /output to the way audio description/captions are provided in virtual reality and beyond. As the long distribution of user need is a significant challenge in creating useful, scalable accessibility applications, we demonstrate how a teachable approach can address these challenges and provide design considerations to help researchers and practitioners working across the accessibility domain.

8 ACKNOWLEDGEMENTS

We would like to acknowledge the engagement of the VICTA and CNIB charities in helping us connect with our citizen designers and user study participants.

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, William 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] Saleema Amershi, James Fogarty, Ashish Kapoor, and Desney Tan. 2009. Designing for End-User Interactive Concept Learning in CueFlik. *Neural Information Processing Systems (NIPS) Workshop on Analysis and Design of Algorithms for Interactive Machine Learning*.

[4] C. Andrews. 2014. Accessible Participatory Design: Engaging and Including Visually Impaired Participants. In *Inclusive Designing.* Springer International Publishing, 201–210.

[5] Harshadha Balasubramanian, Cecily Morrison, Martin Grayson, Zhanat Makhataeva, Rita Marques, Thomas Gable, Dalya Perez, and Edward Cutrell. 2023. Enable Blind Users' Experience in 3D Virtual Environments: The Scene Weaver Prototype. In *In 2023 CHI EA Conference on Human Factors in Computing Systems*, 1–4.

[6] Michelle Carney, Barron Webster, Irene Alvarado, Kyle Phillips, Noura Howell, Jordan Griffith, Jonas Jongejan, Amit Pitaru, and Alexander Chen. 2020. Teachable machine: Approachable Web-based tool for exploring machine learning classification. In *The 2020 CHI EA Conference on Human Factors in Computing Systems*, 1–8.

[7] Da Chen, Yuefeng Chen, Yuhong Li, Feng Mao, Yuan He, and Hui Xue. 2021. Self-Supervised Learning for Few-Shot Image Classification. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1745–1749.

[8] Rebecca Fiebrink, Perry R. Cook, and Dan Trueman. 2011. Human model evaluation in interactive supervised learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 147–156.

[9] C. Finn, P. Abbeel, and S. 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.

[10] Stephan Forchhammer, Amar Abu-Ghazaleh, Gisela Metzler, Claus Garbe, and Thomas Eigentler. 2022. Development of an Image Analysis-Based Prognosis Score Using Google's Teachable Machine in Melanoma. *Cancers* 14, 9: 2243.

[11] Steven M. Goodman, Ping Liu, Dhruv Jain, Emma J. McDonnell, Jon E. Froehlich, and Leah Findlater. 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, 2: 1–23.

[12] 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.

[13] 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.

[14] Hernisa Kacorri. 2017. Teachable machines for accessibility. *ACM SIGACCESS Accessibility and Computing* 119: 10–18.

[15] 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.

[16] Rie, Kamikubo, Kyungjun Lee, and Hernisa Kacorri. 2023. Contributing to Accessibility Datasets: Reflections on Sharing Study Data by Blind People . In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–18.

[17] 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*, 251–254.

[18] Todd Kulesza, Weng-Keen Wong, Simone Stumpf, Stephen Perona, Rachel White, Margaret M. Burnett, Ian Oberst, and Amy J. Ko. 2009. Fixing the program my computer learned. In *Proceedings of the 14th international conference on Intelligent user interfaces*, 187–196.

[19] Brenden M Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B Tenenbaum. 2011. One shot learning of simple visual concepts. *Cognitive Science* 33, 2568 – 2573.

[20] Kyungjun Lee, Abhinav Shrivastava, and Hernisa Kacorri. 2020. Hand-Priming in Object Localization for Assistive Egocentric Vision. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 3422-3432.

[21] Kelly Mack, Emma McDonnell, Dhruv Jain, Lucy Lu Wang, Jon E. Froehlich, and Leah Findlater. 2021. What Do We Mean by "Accessibility Research"? In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–18.

[22] 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.

[23] 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.

[24] Eunbyung Park and Junier B Oliva. 2019. Meta-Curvature. In *Advances in Neural Information Processing Systems* 32, 1-11.

[25] 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.

[26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115, 211-252.

[27] 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.

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

[29] Jonathan Silvertown. 2009. A new dawn for citizen science. *Trends in Ecology & Evolution* 24, 9: 467–471.

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

[31] Mingxing Tan and Quoc Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the ICML International Conference on Machine Learning*, 6105–6114.

[32] Anja Thieme, Cynthia L. Bennett, Cecily Morrison, Edward Cutrell, and Alex S. Taylor. 2018. "I can do everything but see!--How People with Vision Impairments Negotiate their Abilities in Social Contexts. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 203.

[33] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems*, 29, 1-9.

[34] Michele A Williams, Amy Hurst, and Shaun K Kane. 2013. "Pray Before You Step out": Describing Personal and Situational Blind Navigation Behaviors. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility*: 1-8.

[35] Yuhang Zhao, Edward Cutrell, Christian Holz, Meredith Ringel Morris, Eyal Ofek, and Andrew D. Wilson. 2019. SeeingVR. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–14.

[36] Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. 2019. Fast Context Adaptation via Meta-Learning. In *Proceedings of the ICML Conference on Machine Learning*, 7693–7702.