ReCog: Supporting Blind People in Recognizing Personal Objects

Dragan Ahmetovic
Università degli Studi di Milano
dragan.ahmetovic@unimi.it

Daisuke Sato
Carnegie Mellon University, IBM
daisukes@cmu.edu

Uran Oh
Ewha Womans University
uran.oh@ewha.ac.kr

Tatsuya Ishihara
IBM Research - Tokyo
tisihara@jp.ibm.com

Kris Kitani
Carnegie Mellon University
kkitani@cs.cmu.edu

Chieko Asakawa
Carnegie Mellon University, IBM
chiekoa@cs.cmu.edu

ABSTRACT

We present ReCog, a mobile app that enables blind users to recognize objects by training a deep network with their own photos of such objects. This functionality is useful to differentiate personal objects, which cannot be recognized with pre-trained recognizers and may lack distinguishing tactile features. To ensure that the objects are well-framed in the captured photos, ReCog integrates a camera-aiming guidance that tracks target objects and instructs the user through verbal and sonification feedback to appropriately frame them.

We report a two-session study with 10 blind participants using ReCog for object training and recognition, with and without guidance. We show that ReCog enables blind users to train and recognize their personal objects, and that camera-aiming guidance helps novice users to increase their confidence, achieve better accuracy, and learn strategies to capture better photos.

Author Keywords

Visual impairment; object recognition; photography guidance.

CCS Concepts

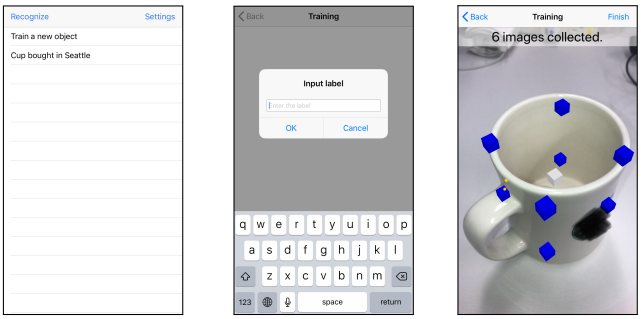
•**Social and professional topics** → *Assistive technologies;*
•**Computing methodologies** → *Mixed / augmented reality;*
•**Human-centered computing** → *Auditory feedback;*

INTRODUCTION

To support blind users in recognizing objects, novel assistive technologies use machine learning and computer vision, trained on large image datasets. Mobile applications such as Microsoft Seeing AI [24] are trained to recognize common objects (*e.g.*, car, cup) or commercial products (*e.g.*, Coke, Pepsi). However, for personal objects such as clothing, handmade items, local products, or pictures of loved ones, a general purpose recognizer cannot be used.

CHI'20, April 25–30, 2020, Honolulu, HI, USA.

ACM ISBN 978-1-4503-6708-0/20/04. . . $15.00.
http://dx.doi.org/10.1145/3313831.3376143



(a) Main screen (b) Label input (c) Photo taking

Figure 1: ReCog app screens

To address this issue, we present ReCog, a mobile app that enables blind users to capture photos of their personal objects, and use them to train a deep neural network that can recognize such objects (see Figure 1). Focusing the camera to capture photos is difficult for blind people [15], and results in photos of inconsistent quality. Instead, consistent photos are desirable because they improve recognition accuracy [17]. Thus, besides manual photo capturing, ReCog also provides camera-aiming guidance to track the objects in the camera frame, using vocal feedback and sonification to guide the user to position the camera with respect to the object.

We evaluated the system through a study with 10 blind participants across two sessions. The first session explored photo capturing with and without guidance in a controlled scenario, and collected participants' subjective feedback. The second session studied in-the-wild usage of the system during the following 3 days. A final questionnaire assessed changes in the participants' opinions of the system after prolonged usage. The participants perceived the system to be usable, and were able to use it autonomously. Most of them had limited knowledge on how to aim a camera, so they preferred using guidance to capture photos, which also improved the image quality and recognition accuracy. After prolonged use, participants acquired confidence in their photo capturing skills and shifted away from guidance. However, this interaction modality was still considered better for novice users, and during the training task to ensure that the resulting recognition model is reliable.

RELATED WORK

Blind individuals resort to various strategies to recognize objects that are not easy to identify without sight. Besides memorizing where the objects are placed, which is cognitively demanding, it is possible to use braille labels [12]. These are time-consuming to emboss, especially for a large number of objects, and they are useful only to proficient braille readers, which are a minority [26]. Electronic markers, such as Near-Field Communication (NFC) tags [28], barcodes [34] which are available on most products, or inexpensive visual labels [14, 30] are also used. These can be detected with a smartphone, but may be hard to apply and locate on an object.

Machine learning approaches, trained on photos of objects to recognize, do not require special markings. General purpose recognizers trained on large datasets can recognize thousands of different objects [9, 24]. However, systems using generic training data can only provide high-level object identification (*e.g.*, a bottle). Instead, more detailed descriptions (*e.g.*, mug with flowers) or intra-class discrimination (*e.g.*, Coke vs. Pepsi), may be needed for blind users. Furthermore, generic recognizers cannot detect objects for which they have not been trained or which do not belong to the public domain, such as handmade or personalized objects, local products, or clothes.

Human powered approaches submit photos captured by blind users to a crowdsourcing platform [6] or connect users to sighted assistants [4], which can provide fine-grained recognition without prior labeling or training. However, the response time and quality depend on connection quality, worker availability and knowledge. Crowdsourcing services also have a cost and raise privacy concerns when personal objects are involved, which may not be solved by masking image parts [19].

Personal object recognition [17, 2] is a novel paradigm, that enables blind users to recognize their personal objects, using photos of such objects, captured by the users themselves, to train their own recognizers and perform object recognition. This way it is possible to distinguish between similar objects that cannot be recognized with general recognizers. We designed and implemented ReCog, the first smartphone-based personal object recognizer that guides blind users to autonomously capture photos of their objects of interest and train a personalized deep network with these photos.

Object recognition is more accurate when training and recognition photos are consistent [17]. However, capturing well-framed photos is challenging for blind people [5]. Prior works provide camera aiming trough sonification [36], verbal [38] or vibro-haptic cues [3], using computer vision [15] or crowdsourcing [6] to locate the target. These approaches focus on photo composition [3, 15, 38] or only address object centering but not proximity [36], without specifically aiming for capturing well framed photos for object recognition. Indeed, existing object recognizers do not provide guidance [2], and therefore depend on user's photo capturing ability. Hand detection to support camera aiming for object recognition has been proposed [23], but still heavily relies on user's ability. Instead, our system improves object framing through a robust guidance based on visual-inertial odometry [16] which conveys the distance and direction of the camera from the target object.

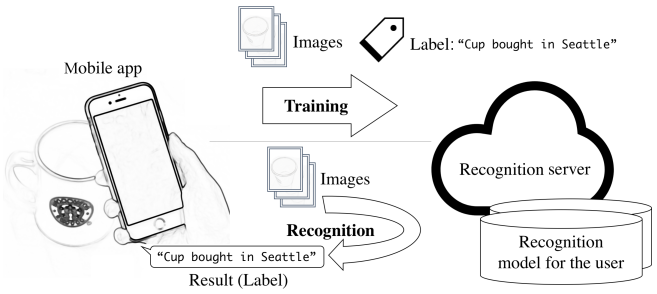


Figure 2: Overview of Personal Object Recognizer (ReCog)

PERSONAL OBJECT RECOGNIZER

ReCog enables blind users to identify and differentiate between their personal objects, which often lack distinguishing tactile features, such as clothing, handmade items, local products, or pictures of loved ones. The system is first trained by the users with their own photos and labels of their personal objects, and it is then able to recognize such objects. The system (see Figure 2) consists in the ReCog mobile app, which runs on the user's smartphone and provides photo capturing functionalities, and the ReCog recognition server, which hosts the recognition model and performs training and recognition.

ReCog Mobile App

The app exposes 2 main activities: *Training* and *Recognition*. The *Training* activity is accessed from the main application view by selecting "Train a new object" (see Figure 1a). In the current version of the app the user is prompted to manually provide a label for the new object (see Figure 1b). This approach is useful when the object label is known, or in presence of sighted assistants that can provide the initial label. If the user cannot identify the object in the first place, other labelling methods, such as friendsourcing [27], or crowdsourcing [35] will be included as a future work.

If manual photo capturing is selected, the user can capture photos of an object by tapping on the touchscreen. ReCog also provides **camera-aiming guidance** (see Figure 1c), which is described in the following section. In both cases, the user is advised to use a plain background surface to improve recognition accuracy. Once enough photos of the object are captured (default 10), the app instructs the user to turn the object to the other side and capture additional photos or to finish the photo capturing if all sides of the object were already captured. The ReCog server starts the recognition model training once the photo capturing is finished and notifies the mobile app through push notifications when the training is completed.

Selecting "Recognize" in the main application view (see Figure 1a) starts the *Recognition* procedure. The photo capturing interface is the same one used for the training, and it can also be performed on single or multiple images (default 5), as defined in the application settings. Recognizing an object on multiple images aims to increase the robustness of the recognition, but it also entails a greater workload from the user. Once captured, the photos are submitted to the ReCog server for recognition. If the object is recognized with a high confidence, the object label is read and the user can recognize a new object.

ReCog Recognition Server

The ReCog recognition server uses a state-of-the-art deep convolutional neural network (CNN) algorithm to train the recognition models using photos captured by the users, and to recognize the captured photos with the trained models. The presented approach is not novel, but it is a central part of our system and therefore we describe it in details. Deep CNN architectures [22] are structured as a cascade of layers, with bottom layers applying the convolution operation on the input image to extract low-level local visual features, while top layers connect these features to characterize different objects. We used the **Inception-v3** architecture to balance between the recognition accuracy and speed [31], implemented on top of the Tensorflow machine learning framework [1].

While deep CNN are currently the state-of-the-art for accurate object recognition, they generally need to be trained on large images datasets. However, it is unfeasible for blind users to capture hundreds of photos for each object for training purposes. We address this issue in two ways: 1) We artificially modify existing images, changing object position, rotation, size, luminosity or randomly erasing image portions [39]. This procedure significantly increases the recognition accuracy, but it also increases the training time. 2) It is possible to reduce training time by training only the top layer with the actual dataset, while low layers, which recognize generalizable image features, are reused from a pre-trained network. This process, called transfer learning [11], achieves reasonable recognition accuracy and the training requires only a fraction of the time needed for full network training.

To achieve high accuracy, but still enable the recognition as soon as possible, we train two different recognition models. The first model (**quick model**) trained through transfer learning for 100 epochs (training iterations), provides object recognition capabilities after only about 10 min. of training. The second model (**full model**) trains the full network for 1000 epochs, which results in a training time of about 3 hours, during which the recognition accuracy gradually improves until the accuracy is saturated. For both models we use Adam [20] optimization algorithm, which is characterized by quick and robust learning. The number of epochs has been defined empirically to balance between accuracy and training speed. The batch size, that is the number of training examples used in each iteration, is set to 32 for best performance. The learning rate, which defines how fast the network learns from the training data, is initially set to 2×10^{-4} in order to yield meaningful results quickly, and it is gradually decreased during training to allow finely grained improvements. These parameters are also selected empirically, based on prior experience with our data.

CAMERA-AIMING GUIDANCE

We designed our guidance approach to ensure that the target object is consistently well framed in the captured photos. This goal was informed by prior work which show that consistent object framing in training and testing photos improves recognition accuracy [17]. To achieve this we track the object within the camera frame, we measure its framing quality through two novel metrics, and we use those metrics to instruct the user to correctly position the camera with respect to the object.

(a) Identifying object position (b) Estimating object size

Figure 3: Photo capturing with camera-aiming guidance

Object Tracking and Pose Estimation

To calculate and track the object position we use ARKit[1], an iOS framework for positional tracking and scene understanding using video camera stream and inertial measurement unit (IMU) sensor values. ARKit uses Simultaneous Localization and Mapping (SLAM), an incremental, online Structure-from-Motion algorithm to map coordinates of the objects in the camera frame to real world coordinates. SLAM concurrently estimates the camera orientation and location in 3D and constructs a sparse 3D point cloud of the environment [8]. For this, SLAM relies on an initialization step which is critical for the stability of the object tracking. The initialization step is best carried out with a purely translational motion, parallel to the scene [25]. The user is instructed with an audio message to face the smartphone camera towards the target object and move the phone from left to right until an audio signal is heard.

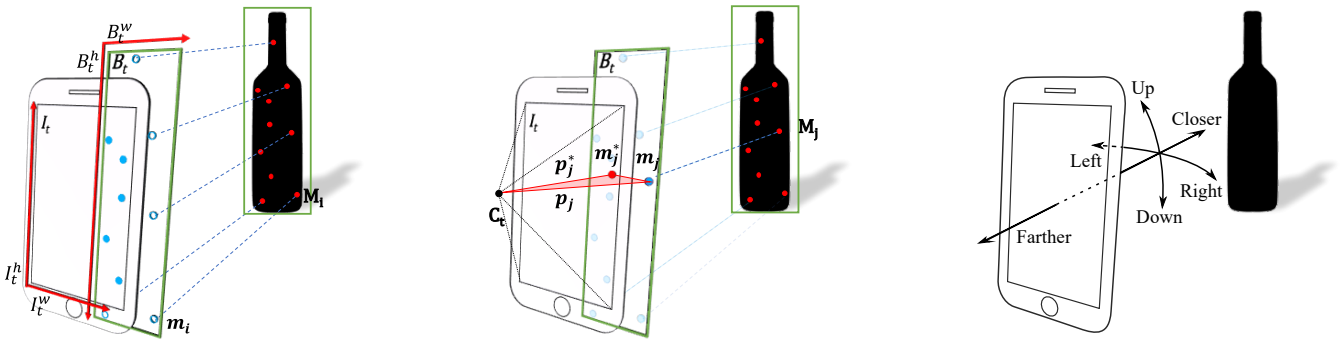
After the initialization, the system has to identify the position of the target object in 3D. First, the system asks the user to slowly move the phone towards the object, while touching the object with the other hand (see Figure 3). During this motion, the system estimates the phone trajectory in 3D. The trajectory, which is fitted to a straight line, is tested for intersection against the point cloud of the target object. The intersection closest to the user is recorded as the point $\boldsymbol{M}^*$. Alternatively, the user can also touch the object with the phone and tap the screen, as in EasySnap in object mode [15].

Then, the system determines the approximate size of the object similarly to [18, 33]. Given the assumption that most common objects have a convex shape, the system starts by inserting the triangle described by the point $\boldsymbol{M}^*$ plus its two closest points in the point cloud to the object point set $\mathscr{M}$. Then, the closest points to $\mathscr{M}$ from the point cloud are incrementally considered by forming a triangle and testing the resulting mesh for convexity. If convexity is maintained, the point is added to $\mathscr{M}$ and the process continues until no more points can be added to the set.

Object Framing Quality Metrics

In contrast with prior work, our approach not only considers the position of the object within the camera frame but also its proximity. To measure the framing quality we define two novel camera pose quality metrics: 1) the *proximity score*, and 2) the *center offset score*.

[1] https://developer.apple.com/arkit/



(a) Proximity Score (b) Center Offset Score (c) Speech feedback

Figure 4: Camera-aiming guidance

Proximity Score

The proximity score measures the distance between the camera and the object. It approaches 1 if the object's size is close to the image size and it is close to 0 when the object in the image is too small. Formally, we consider the current image frame $\boldsymbol{I}_t$, with height h and width w. The intrinsic camera parameter matrix $\boldsymbol{K}_t$ is assumed to be known and static. We estimate the extrinsic camera parameters $\boldsymbol{P}_t = [\boldsymbol{R}_t|\boldsymbol{t}_t]$, where $\boldsymbol{R}_t$ and $\boldsymbol{t}_t$ are respectively a rotation matrix and a translation vector. Each 3D point $\boldsymbol{M}_i \in \mathscr{M}$ in world coordinates is projected to a 2D point $\boldsymbol{m}_i = \boldsymbol{K}_t\boldsymbol{P}_t\boldsymbol{M}_i$ in camera coordinates based on [13] (see Figure 4a). The object's minimum bounding rectangle $\boldsymbol{B}_t$, with height h_{B_t} and width w_{B_t} is computed from the 2D projected points $\boldsymbol{m}_i$. Then, we compute the *Proximity score*:

$$D_t = \max\left(\frac{h_{B_t}}{h}, \frac{w_{B_t}}{w}\right)$$

Center Offset Score

The center offset score measures how centered the object is in the camera field of view (see Figure 4b). It is close to 1 when the object is centered and approaches 0 when the object is out of the field of view. A set m^- is created considering the object points $\boldsymbol{m}_i$ that lie outside of the image $\boldsymbol{I}_t$. Then, each point $\boldsymbol{m}_j \in m^-$, is associated to its closest point $\boldsymbol{m}_j^*$ inside $\boldsymbol{I}_t$. Denoting $\boldsymbol{C}_t = -\boldsymbol{R}_t^{-1}\boldsymbol{t}_t$ as the camera center, and considering the vectors $\boldsymbol{p}_j = \boldsymbol{C}_t\boldsymbol{m}_j$, $\boldsymbol{p}_j^* = \boldsymbol{C}_t\boldsymbol{m}_j^*$, the *Center offset score* is:

$$T_t = \arg\min_j \left(\frac{\boldsymbol{p}_j \cdot \boldsymbol{p}_j^*}{\|\boldsymbol{p}_j\|\|\boldsymbol{p}_j^*\|}\right)$$

Sonification and Speech Feedback

Initially we explored guidance using solely sonification [36] or verbal instructions [15], augmented to also provide distance information. Preliminary tests with 12 blind participants exposed a mild preference for the sonification modality but advantages of both approaches were uncovered: verbal instructions better conveyed direction, while sonification was able to convey the amount of movement needed. Thus, we combined the two modalities into one single interaction technique and tuned its sonification parameters and verbal instructions with the support of a blind accessibility expert from our group.

Sonification

This modality mimics the idea of tuning a string instrument by correctly aiming the camera [37]. Our system generates two sine wave sounds; the first one at a fixed frequency $F_\text{t} = 440$Hz, and the second one at a variable frequency F_v, which changes according to T_t, D_t and a scaling parameter $\alpha = 10$.

$$F_\text{v} = F_\text{t} + \alpha \cdot \begin{cases} T_t & (0.5 \leq D_t \leq 0.8) \\ D_t & (otherwise) \end{cases}$$

As the camera points away from the object, the sound interference causes a pulsating tone which conveys a sense of urgency. To maximize the effect, the sound volume is proportional to the pulse frequency. By adjusting the camera orientation, the pulse caused by the interference softly disappears as the two frequencies become one.

Speech Feedback

This mode provides verbal instructions as in prior works [3, 38]. The system speaks "left", "right", "up", and "down" according to T_t and "closer" and "farther" according to D_t. However, "left", "right", "up", and "down" are not used for panning (camera translation) but for tilting (camera rotation) as shown in Figure 4c. To convey a greater sense of urgency, the voice pitch is proportional to D_t and T_t.

Automatic Photo Collection

Once the object is centered, that is whenever D_t and T_t are within predefined bounds ($0.5 \leq D_t \leq 0.8$, $T_t = 1$), the system automatically takes a photo. This is particularly beneficial as the user does not need to touch the phone for taking photos, which can cause unwanted blur.

In addition to capturing the photo, the system also verifies that the camera position has changed with respect to previously captured photos. This constraint is needed to build a diversified collection of photos, in order to train a better recognizer. To achieve this, the system computes a number of uniformly distributed viewpoints on a hemisphere around the initial object point $\boldsymbol{M}^*$. Whenever a photo is about to be taken, the camera position is projected onto the hemisphere and the closest viewpoint is found. If the viewpoint has not been already recorded, the photo is taken. Otherwise, the user is prompted to move the camera through verbal messages and sonification.

ID	Sex	Age	Visual Impairment		Years of experience with		
			Type	Onset	iPhone	VoiceOver	Camera
P1	M	64	Totally blind	Birth	12 years	8 years	2 years
P2	F	70	Totally blind	Birth	10 years	6 years	1 year
P3	M	70	Light percept.	Birth	3 years	3 years	Rarely
P4	M	42	< 20/400	Birth	2 years	2 years	2 years
P5	F	44	Totally blind	Age 20	3 years	3 years	3 years
P6	M	45	Totally blind	Age 2	15 years	7 years	3 years
P7	F	62	Light percept.	Age 10	10 years	10 years	5 years
P8	M	41	Light percept.	Age 10	10 years	10 years	10 years
P9	F	44	Totally blind	Age 19	8 years	8 years	8 years
P10	M	43	Totally blind	Birth	10 years	8 years	10 years

Table 1: Participant demographic data

(a) PP Roast (b) Decaf (c) Candies (d) Gummies

Figure 5: Four default objects

USER STUDY

To assess ReCog performance and its acceptance, we conducted a user study with 10 blind participants. Specifically, we were interested in addressing the following research questions:

- Does camera-aiming guidance impact object recognition?
- What are participants' opinions of ReCog and the guidance?
- Do their opinions change after familiarizing with ReCog?

The study was divided in two sessions. The first one studied training and recognition activities, with and without guidance in a controlled setting. A follow-up questionnaire collected participants' opinions of the system. The second session took place during the following 3 days. Participants were asked to autonomously train and test additional objects of their choice at home. A final questionnaire assessed how participants' opinion of the system changed after prolonged use.

Participants

We recruited 10 participants (4 female, see Table 1). Six were totally blind, three had light perception, and one was legally blind, with residual sight insufficient to recognize objects. Their average age was 52.5 (SD = 12.3). All of them have used iPhone for at least 2 years (M = 8.3, SD = 4.3), and had more than 2 years of experience with VoiceOver screenreader (M = 6.5, SD = 2.9). Camera expertise was not equally distributed among all participants (M = 4.41 years, SD = 3.7). In particular, one participant had only 1 year of camera usage experience, and one rarely ever used a camera. However, all participants had experience with at least one camera-based object recognition app such as SeeingAI or TapTapSee[32].

Apparatus and Experimental Setting

We implemented ReCog as an iOs app. Experiments were conducted on iPhone 7 devices updated to the latest iOS version at the time of the study (11.2.2). The recognition server was equipped with 128GB of DDR4 RAM, 4 NVIDIA GeForce GTX 1080 Ti video cards used to run the CNN training procedure, and an Intel Xeon E5-2660 v3 2.60GHz CPU with 10 cores and 20 threads used for the recognition. For the experiments we used a well-lit surface covered with a plain colored tablecloth to limit background features that may influence the recognition (See Figure 5). External objects were removed from the area and participants were positioned so that, while pointing the camera towards the object in front of them, there are not many features that could influence the recognition.

Procedure

Before the study we invited the participants to read the help documentation[2], which describes the main functionalities and the interaction with the system. At the beginning of the study, before the first session, participants were provided with the study consent form. We then conducted a tutorial that follows the same indications provided in the help documentation. It involved the training of ReCog with two objects: a *t-shirt* and a *mug*. These two objects were chosen because they require a different interaction during photo capturing. A flat object such as t-shirt requires the smartphone to be held horizontally, parallel to the surface on which the object is placed. A vertically standing object, such as a mug, requires the smartphone to be held vertically, standing on the table and parallel to the object.

First Session

The session was video recorded to study the behavior of the participants during the procedure. Participants were asked to take breaks when desired. The session was 150 min. long, and it consisted of two tasks: object training and recognition. For both tasks, participants were asked to capture photos of a set of 4 predefined objects and 4 objects of their choice. All objects were trained and recognized with and without camera-aiming guidance in a counterbalanced order to offset the learning effects that may influence the performance of the system and the participants' perception of it. Participants with even IDs had camera-aiming guidance as first condition, while others had it as second condition.

Participants began the training task with a practice object (*Chewing Gums*) to verify that they were performing the procedure correctly. If any problem were noted at this stage, the examiners repeated the corresponding help instructions to make sure that the participants performed the training activity correctly. Afterwards, the participants were asked to train the system with four default objects: two coffee k-cups (*Decaf* and *PP Roast*), and two candy boxes (*Candies* and *Gummies*) (see Figure 5). These object pairs were chosen because they are similar in shape, and therefore hard to distinguish non-visually. The participants were then asked to train the system with four of their own objects that they would want to recognize with ReCog. For each object, participants were instructed to repeat the photo capturing for all major sides (*e.g.*, avoiding short sides on flat boxes). After the training in one condition, the task was repeated in the remaining condition.

[2] http://por.bitballoon.com/

After the training, a questionnaire was administered to collect the participants' demographic data and to assess their experience with the system. The questionnaire included the system usability scale questions [7] to measure the overall appreciation of the system, and subjective ratings regarding the system usage with and without guidance. Then, the participants were asked to perform the object recognition task in both conditions and in the same order as for the training task, using the same set of objects. For default objects, participants were asked to perform at least five recognitions, for each side of the object. This was necessary to acquire a sufficient number of photos for the recognition accuracy computation. Note that for each recognition, five photos were captured by default.

Second Session

Participants were asked to continue using the system for other three days, in order to study how their interaction with the app and their opinion of the system evolved after prolonged usage in a more naturalistic scenario. They were instructed to train up to 20 additional objects. The number of objects was limited due to computational constraints, to allow all users to access the system concurrently. Participants could train and test any object of their choice. However, they were notified that the experimenters would review the photos taken, and therefore to avoid private objects. The objects could be trained and tested with or without camera-aiming guidance. Each photo captured by the participants was associated with the corresponding object label and its timestamp. At the end of the second session, the participants were administered the same questionnaire as in the first session, except for the demographic part that has been already collected. This second questionnaire was used to compare the users opinion before and after acquiring experience in natural usage of the system.

RESULTS

We present the results of the user study and evaluation of ReCog. We focus in particular on six aspects: 1) analysis of the collected photo data, 2) evaluation of the photo quality with and without camera-aiming guidance, 3) object recognition accuracy on photos captured by participants, 4) evaluation of the system using the System Usability Scale, 5) evaluation of the camera-aiming guidance, and 6) participants' observations.

Photo Data Analysis

We analyze the photos captured for training and testing.

Training Photos.

During Session 1, the system captures 10 training photos for each side of each object. Thus, for default objects, a total of 60 training photos were captured by each participant: 10 photos for the k-cups and 20 photos for the boxes. This was repeated with and without guidance. Some participants erroneously repeated the training for some of the object sides, and therefore they captured more photos than needed. For a fair comparison, excess data was discarded during the training of the recognition model. The system captured 10 to 40 training photos for each participants' objects, due to the varying number of sides. On average, for the default objects, 82 training photos were captured (SD = 19.9) with guidance, and 76.9 (SD = 20.6) without. A total of 1589 photos were captured.

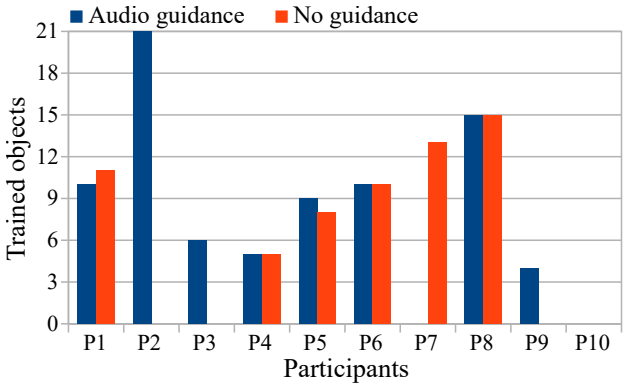


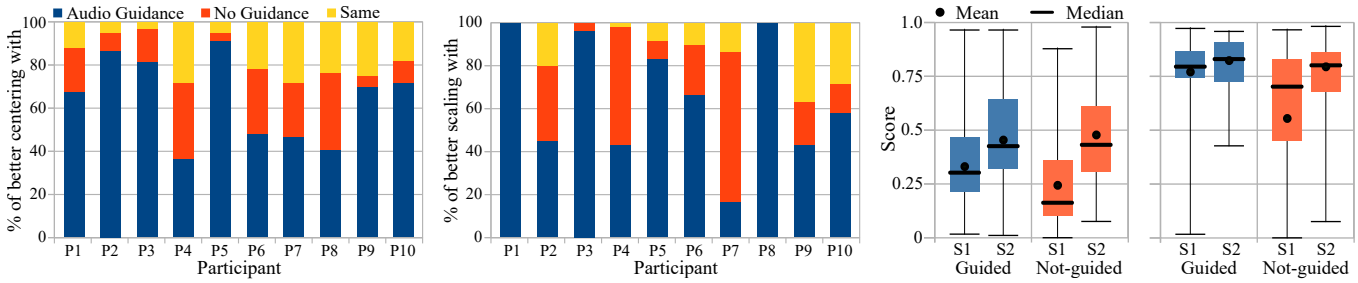
Figure 6: Number of objects trained in Session 2

During Session 2, 142 objects were trained. The number varied noticeably between participants, from 0 to 30 (see Figure 6). On average, 8 (SD = 6.5) objects were trained with guidance and 6.2 objects (SD = 6.0) without. 6 participants balanced between the two modalities, while others used one modality only. $P2$, $P3$ and $P9$ performed the training only with guidance, while $P7$ captured photos only without (see Figure 6). $P10$ captured no additional objects in Session 2, while two participants, $P3$ and $P6$, trained some of their objects multiple times under different labels. $P3$ trained 5 of the objects twice with guidance, while $P6$ trained 2 of the objects twice with guidance and 1 object twice without guidance.

The types of the objects trained by the users in Session 1 and 2 are in substantial agreement with the objects of interest for people with visual impairments identified in prior literature [17]. All participants trained food and drink items that are difficult to recognize otherwise, such as cans, boxes and bags. $P2$, $P5$, $P7$ and $P8$ trained hygiene and cleaning products, while $P2$, $P4$ and $P8$ included t-shirts and other clothing. Female participants ($P2$, $P5$, $P7$ and $P9$) had cosmetic products among their objects of interest, $P4$ and $P5$ also included medicines, and $P1$ and $P3$ added appliances such as remote controllers or radios. In addition to categories found in prior literature, three new object types were also present. Specifically, $P2$, $P7$ and $P8$ also included objects related to pets or guide dogs, such as dog food, treats and wipes. $P9$ and $P10$, who have sighted children, included colouring and story books, while $P3$ included pictures of his family members.

Testing Photos

For the recognition of default objects in Session 1, 5 photos were captured each time. We repeated this procedure 5 times for each side of the object, both with and without guidance. Thus, in total, every participant was supposed to capture 150 photos for each condition. Again, some participants captured more photos which were discarded, and others captured less photos, which was not a problem because having a different number of testing photos has no impact on the recognition model or on the resulting accuracy. Specifically, $P5$ captured 65 photos with guidance and 135 without; $P9$ captured 130 with guidance and 100 without; and $P10$ captured 145 with guidance and 85 without. In total, 1390 testing photos were captured with camera-aiming guidance, and 1370 without.



(a) Centering scores per participant (b) Scaling scores per participant (c) Proximity (d) Center Offset

Figure 7: Photo quality assessment scores

For participants' objects, on average 50.5 testing photos were captured with guidance (SD = 29.9) and 62 without guidance (SD = 40.3) during the first session. During Session 2, similarly to the training photos, also in the case of testing photos there was a high discrepancy between the number of photos per participant and per condition. Indeed, with camera-aiming guidance, the average number of photos captured was 83 (SD = 103.3), while without guidance it was 46 (SD = 96.7).

Most participants trained the system with objects that they never attempted to recognize ($P1$, $P4$, $P5$, $P6$, $P7$). Indeed, the total number of tested objects was 89, much lower than the 142 trained objects. The average number of tested object with guidance was 3.7 (SD = 4.79) and 5.2 (SD = 7.16) without. $P5$ and $P6$, also tried to recognize objects that were not trained. $P5$ tried to recognize one untrained object while $P6$ tried to recognize 7 untrained objects (all k-cups of different flavours) in both interaction modalities. On this matter, $P6$ explained:

> *"Without the app being able to say that the object was not trained, its practical use is greatly limited."*

Photo Quality Assessment

To evaluate the impact of the camera-aiming guidance on the quality of the photos, two annotators labeled and compared the training photos of the default objects with and without guidance collected in Session 1. The annotators compared the photos in pairs, one from each condition, in the same capturing order (i.e., first photo captured with guidance is compared to the first one captured without). The metrics used were derived from the camera pose quality metrics defined previously:

> **centering**: the annotators selected the photo in which the distance between the object center and the center of the photo appeared to be shorter, if not equal.
>
> **scaling**: the annotators selected the photo in which the object size appeared to be closer to the size of the photo, without being too big or too small, if not equal.

Cohen's Kappa coefficient was computed to measure the inter-rater agreement on 20% of the data which was labeled by both annotators [10]. The annotators have assigned the same scores on 72.5% of the data for the centering metric, reaching a moderate agreement score of 0.58. For scaling, the identical scores were assigned in 93.3% of the cases, with a substantial agreement score of 0.76.

As shown in Figure 7a, 64.2% of photos were better with guidance (SD = 19.8%) and 18.8% without (SD = 12.2%). Wilcoxon Signed Rank Test revealed that the difference is significant ($Z = 3.78$, $p < .001$). Thus, **participants were better at centering the objects with camera-aiming guidance** than without. Similarly, as shown in Figure 7b, **photos captured with camera-aiming guidance received significantly higher scaling score** (M = 65.3%, SD = 28.9%) than without guidance (M = 22.9%, SD = 23.9%); $Z = 2.80$, $p < .01$.

To evaluate how photo quality changed between sessions we could not use the same metrics because the same objects were not present in both sessions and different conditions. Thus, we manually labeled $1/6$ of Session 1 photos (100 images) for each condition and the same number of Session 2 photos. On these, we measured raw proximity and center offset scores and compared them across conditions. Between Session 1 and Session 2 the proximity metric improved from 0.35 to 0.48 on average for the guided condition and from 0.25 to 0.49 for the not guided condition (see Figure 7c). In both cases the difference was statistically significant based on Mann-Whitney U test ($p < .001$). Instead, the center offset metric improved from 0.77 to 0.80 for the guided condition and from 0.59 to 0.78 for the non-guided condition (see Figure 7d). Only the latter difference was statistically significant ($p < .001$). Thus, **with prolonged usage the scores improve for both conditions and in particular for non-guided condition**.

Recognition Accuracy

Since the photos captured with guidance are better centered and scaled, we hypothesize that:

> **H1: The photos captured with camera-aiming guidance are recognized more accurately by the system**.

Since the system selects the recognition result with the highest confidence score of 5 testing photos, we also expect that:

> **H2: Calculating the recognition over multiple photos improves the recognition accuracy**.

To test these hypotheses, we ran the experiments with the photos of default objects from Session 1. We conducted a $2\times2\times2$ ANOVA considering as factors the guidance mode (with vs. without) and the number of testing photos (1 vs. 5). In addition, we also included the training mode (quick vs. full) as a factor since it is also expected to affect the accuracy.

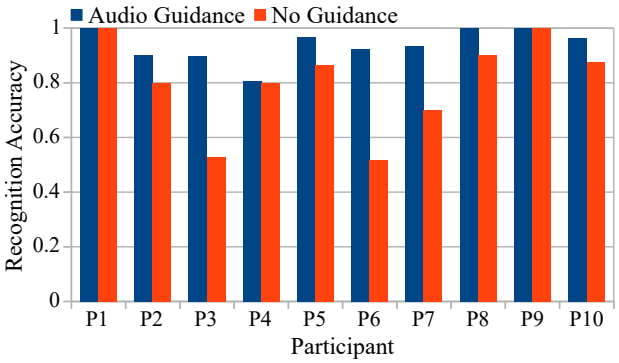


Figure 8: Full training accuracy with and without guidance

The results revealed that there is a significant interaction effect between guidance mode and training mode; $F_{(1,9)} = 7$, $p < .01$. As expected, the post-hoc analysis with paired t-tests showed that the recognition accuracy with full training mode was consistently higher than with quick training, both with guidance ($p < .001$) and without guidance ($p < .05$). However, the effect of the guidance mode within the same training mode differed. Indeed, the average accuracy with camera-aiming guidance ($M = 0.94$ $SD = 0.06$) was significantly higher than without guidance ($M = 0.83$, $SD = 0.17$) for full training ($t_9 = 2.61$, $p < .05$); see Figure 8.

Conversely, there was no significant difference in recognition accuracy between the two guidance modes for quick training modality (with guidance: $M = 0.64$, $SD = 0.14$; without guidance: $M = 0.68$, $SD = 0.19$). Thus we conclude that **H1 is true, but only for the full training mode**. Instead, the main effect of the number of testing photos and other interaction effects were not found to be significant. Therefore, **there is no effect of the number of testing photos on recognition accuracy (H2)**.

As previously noted, participants' objects were extremely different in type, material, number of sides and overall complexity. Due to this, a fair comparison of the accuracy in recognizing the objects in Session 2 could not be conducted. Nonetheless, we note that for most participants the accuracy in the recognition of their own objects was consistent with the results obtained for the default objects in Session 1. For example, for P1 the average accuracy with guidance was 0.95 and 0.86 without guidance. Similar results were obtained for P6 and P8. However, accuracy scores were lower in presence of transparent (P2: 0.6 guided) or almost identical objects (P3: 0.68 guided, 3 of the 6 objects were very similar remote controls).

System Usability Scale

We also collected the participants' subjective evaluation of the system after each session in the form of System Usability Scale [7] and additional Likert scale questions. At the end of the first session, we collected the answers in person, while at the end of the second one, we collected them via e-mail. Two participants ($P9$ and $P10$) never replied regarding the second questionnaire, and therefore they were omitted from the following analysis.

The results, shown in Figure 9, report an average score of 81.9 in the first session ($SD = 5.8$), and 72.2 in the second ($SD = 15.7$). While both scores can be considered good [29], there was a noticeable decrease after prolonged usage. Such a decrease is actually expected as users acquire experience with a system [21].

Considering specific questions, we note that the first question, which assesses how frequently the participants would use the system, decreased from an average of 4 ($SD = 0.6$) to 2.5 ($SD = 1.3$). The decrease was found to be statistically significant ($p < .01$) with a paired t-test. Question 6, which assessed the perceived inconsistency of the system also increased significantly ($p < .05$), from 1.4 ($SD = 0.5$) to 2.0 ($SD = 0.9$). No other answers were found to vary significantly.

Camera-Aiming Guidance Evaluation

Participants were asked to rate the photo capturing functionality with and without guidance in terms of the perceived ease of use, efficiency and confidence on a Likert-like scale ranging from 1 to 7. As shown in Figure 10a, the participants perceived the guidance to be easier to use, more efficient and they felt more confident using the system with it. In particular, $P2$ reported that she did not feel confident using the app without guidance. Indeed, this participant favoured guidance considering all metrics (efficiency, precision, ease of use), different levels of expertise (novices and experts), and both training and recognition tasks. This is also confirmed by the photo data analysis (see Figure 6), which shows that this participant did not train any objects without guidance.

However, we also found a significant increase in users' confidence in using the app without guidance after the second session (t-test $p < .01$). Indeed, the average confidence score using the app without guidance increased from 3.9 ($SD = 1.97$) to 5.14 ($SD = 2.03$). Our intuition is that the participants gradually learned to aim the camera with guidance and therefore became more capable in interacting with the app even without it. This claim was supported by participants' own remarks. For example, $P1$ reported:

> *"Because I did with the audio first, I sort of knew how to center it, but I didn't know if I succeeded."*

This finding is further confirmed by the changes in the participants' preferred photo capturing modality between Session 1 and Session 2 (see Figure 10b). In particular, $P7$, who already had experience in capturing photos without guidance, expressed a strong preference for this interaction modality. Indeed this participant trained Session 2 objects without guidance only, as we can see in the photo data analysis (see Figure 6).

While all participants agreed in both sessions that novice users could benefit the most from the guidance, after the second session, the percentage of those who considered that the preferred interaction for expert users would be without guidance increased from 30% to 75% (Mc'Nemar's test $p < .05$). No other scores changed significantly. As a result of the increased confidence, after the second session, 37.5% of the participants reported that they would prefer to interact with the app without guidance during the recognition.

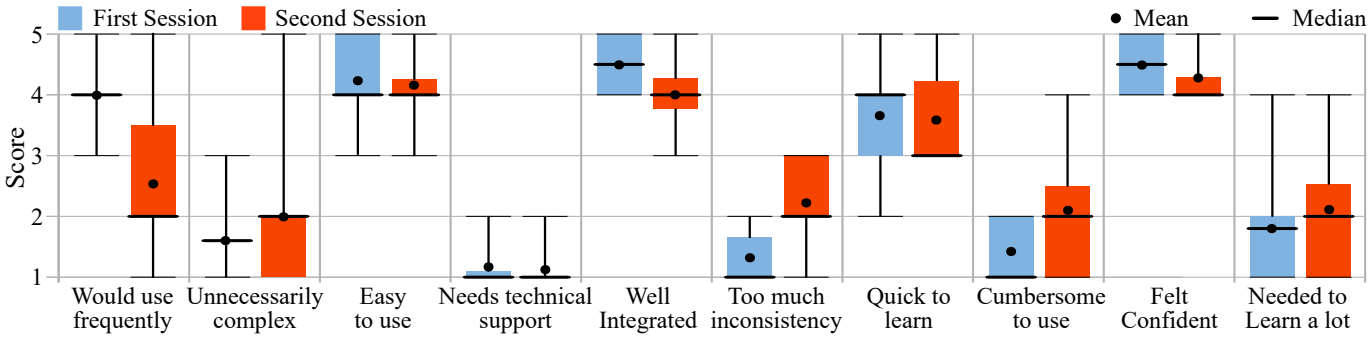


Figure 9: System Usability Scale scores

Nonetheless, the participants unanimously stated that they would still rely on guidance during the training. Even *P*7, who only used ReCog without guidance, agreed that guidance would be useful for novice users and during training in order to create reliable recognition models (see Figure 10b):

> *"I like this (guidance) for training, but not for recognition... because I was getting feedback I felt like I was doing it correctly and gathering the information it was supposed to gather"*

Participants' Observations

The participants also reported observations about the system that they uncovered with prolonged usage.

Motivation and Use Cases.
ReCog does not intend to be an all-purpose object recognizer. Instead, it targets those objects that cannot be recognized otherwise. Some of the participants' objects (clothing, pictures, medicines) belonged to this category, and indeed, participants appreciated this capability of the system (*P*1, *P*4, *P*5, *P*5, *P*8). In particular, *P*8 says:

> *"I began to see how an object recognition software like yours could help me in everyday life. It is an alternative solution to recognize things that do not have bar codes or the bar codes not being on the database"*

End-User Training Is Required.
One of the main concerns that the participants had is the effort needed to collect the photos for training the recognition model (*P*1, *P*2, *P*4, *P*5, *P*7, *P*8). For example, *P*8 remarked:

> *"I don't have to spend time to train other recognizers."*

Labeling Unknown Objects.
Another difficulty reported by the participants (*P*1, *P*2, *P*4, *P*5, *P*8), and highlighted in the prior literature [17] is labeling, which requires knowing what an object is before training. One possible solution is labeling objects immediately after receiving them, while still knowing what they are. However, that was also considered a limitation by *P*2:

> *"I don't believe that my life is always structured enough to do the training when I still know what the items are."*

To mitigate this issue, *P*3 and *P*7 suggested to use crowdsourcing [6] or video assistance [4] for labelling.

Recognition on Multiple Photos.
To improve the recognition accuracy, we included the possibility to capture and perform the recognition using multiple photos, returning the highest confidence score among the detected objects. For this study, we set the number of photos to 5. This feature, however, was perceived negatively by the participants. Notably, *P*7 states:

> *"I would cut down on the amount of pictures needed to recognize an object."*

Difficulties in Tracking Trained Objects.
As we have seen, many participants trained the system with objects that they never attempted to recognize afterwards. This may mean that the participants trained the objects that they thought they might need, but that they did not need all of them during the limited duration of the study.

However, others also tried to recognize objects that were not trained, or trained the same objects multiple times. This suggests that it is difficult for users to keep track of trained objects once their number increases, particularly if those objects are of the same type, such as k-cups trained by *P*6. This may lead to the same objects being trained multiple times, users trying to recognize objects that were never trained, or not trying to recognize previously trained objects.

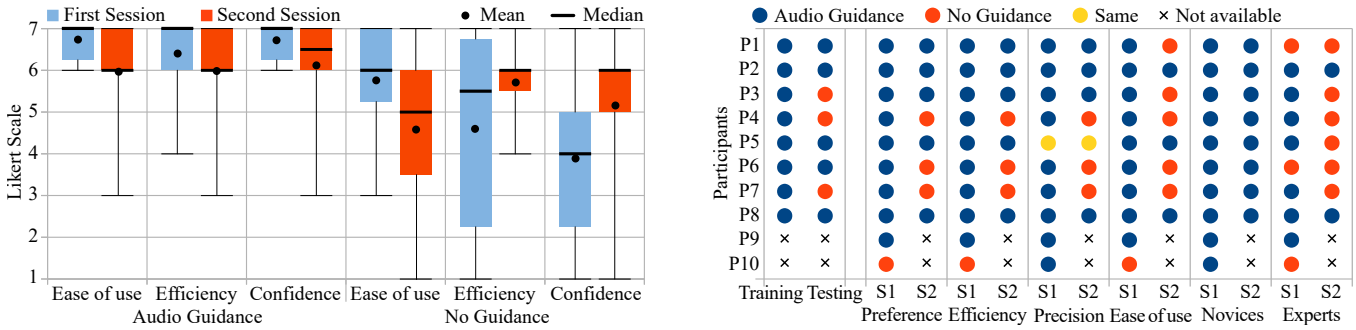
DISCUSSION

We discuss the results of the evaluation of ReCog.

Photo Quality and Its Impact on Accuracy

As prior research hints [17], consistency between training and testing photos may improve the recognition accuracy. On the one hand, improvements in computer vision can mitigate some of the real-world sources of inconsistency, such as luminosity variations. On the other hand, user guidance can be effective for improving consistency in terms of object framing and scaling in the captured photos [23].

We confirm this finding, and show how our technique improves object framing consistency through audio-driven camera-aiming guidance, which also results in a higher recognition accuracy. Our approach for enforcing consistency in the captured photos could also be used to improve the recognition accuracy if the training was performed by others. This could be useful in the case of a generic object recognizer, or when submitting an image query to a crowdsourcing service.



(a) Qualitative scores for camera-aiming guidance

(b) Participants' photo capturing preferences

Figure 10: camera-aiming guidance qualitative evaluation and usage preferences

The Effectiveness of Full Training and Multi-Photo Testing

Results suggest that full training achieves the high recognition accuracy and benefits most from the increased quality of the captured photos using guidance. Nonetheless, quick training provides results much faster, and therefore can still be useful for providing intermediate but immediate recognition results.

Surprisingly, capturing more testing photos does not improve the accuracy. Instead, it seems that the photos captured in sequence have similar visual characteristics and therefore produce the same results; if the recognition is accurate on one photo it will likely be accurate also on others in the same sequence. Since the use of multiple test photos does not improve the accuracy and entails a higher workload for the users, it will be removed from the future iterations of the system.

ReCog as a Learning Tool for Photo Capturing

Participants remarked that they had limited knowledge on how to capture well-framed photos (*P*2, *P*6, *P*7, *P*8), in particular regarding camera distance. Thus, participating to the study was a valuable source of insight. For example, *P*6 commented:

> *"Participation to research was most educational. How far the camera has to be, how sensitive it is to tilt. I had no clue. I thought the closer the camera is the better the image is going to be. Too close it's blurry and you don't get the whole picture. And light and background and all these things I learned it through your research. For example I didn't know that light can make difference."*

This is confirmed by photo quality improvement in Session 2 as well as qualitative scores and preferences (see Figures 7 and 10). Indeed, several participants found it easier to use the system without guidance after having used it with guidance first (*P*1, *P*3, *P*4, *P*6, *P*7). Furthermore all participants agreed that guidance would be useful for novices, and 6 out of 8 stated that for experts interacting without guidance would be better.

Thus, ReCog improves the photo capturing skills of the users by teaching them how their camera aiming impacts the quality of the captured photos. We believe that, through prolonged usage of the system with audio guidance first and without afterwards, blind users may acquire the knowledge on how to capture good photos, which would be beneficial for other similar software or photo sharing on social media [5].

CONCLUSION

We designed and developed ReCog, an interactive smartphone application that enables blind people to recognize their personal objects. This is achieved by capturing photos of such objects and training a recognition model with them. This approach complements general object recognizers, which can recognize common objects only, and crowdsourcing approaches, which rely on human intervention for detailed object recognition. Since capturing well-framed photos is a known difficulty for blind people, we augmented our system with a camera-aiming guidance module that supports the users while capturing photos.

We evaluated the system with 10 blind participants who found it to be usable and accurate. During Session 1, we uncovered a subjective preference for camera-aiming guidance. The analysis of the captured photos confirmed that the use of the guidance modality results in more consistent photo capturing. In particular, the captured objects are better centered and scaled with respect to the photo frame.

During Session 2, the participants gradually started to prefer to use the system without camera-aiming guidance as they improved and acquired confidence in their photo taking skills through prolonged usage of the system. However they still felt more confident using the system with camera-aiming guidance for the training of the object recognizer to ensure a higher recognition accuracy.

We also discovered specific limitations due to the nature of the object recognition technology. While single photos are sufficient at testing time, multiple photos of an object need to be captured during the training, which users may find cumbersome. The system relies on the labeling of the trained objects, which may require sighted assistance or the using an external system. As a future work, we will integrate ReCog with general object recognizers and crowdsourcing approaches in order to minimize the user's need for intervention.

ACKNOWLEDGEMENTS

We would like to thank all the participants who took part in our user study. This work was sponsored in part by Shimizu Corporation and Uptake (Carnegie Mellon University Machine Learning for Social Good fund).

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. USENIX Association.

[2] Envision AI. Accessed: 2019-12-29. `https://www.letsenvision.com`

[3] Jan Balata, Zdenek Mikovec, and Lukas Neoproud. 2015. BlindCamera: Central and Golden-ratio Composition for Blind Photographers. In *Proceedings of the Mulitimedia, Interaction, Design and Innnovation (MIDI '15)*. ACM, 8:1–8:8.

[4] BeMyEyes. Accessed: 2019-12-29. `http://www.bemyeyes.org`

[5] Cynthia L Bennett, Martez E Mott, Edward Cutrell, Meredith Ringel Morris, and others. 2018. How teens with visual impairments take, edit, and share photos on social media. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 76.

[6] Jeffrey P Bigham, Chandrika Jayant, Andrew Miller, Brandyn White, and Tom Yeh. 2010. VizWiz:: LocateIt-enabling blind people to locate objects in their environment. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. IEEE, 65–72.

[7] John Brooke and others. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194, 4–7.

[8] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. 2016. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics* 32, 6, 1309–1332.

[9] CamFind. Accessed: 2019-12-29. `http://www.camfindapp.com`

[10] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20, 1, 37–46.

[11] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2014. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In *Proceedings of the 31st International Conference on Machine Learning*, Vol. 32. PMLR, Bejing, China, 647–655.

[12] John A Gardner. 1996. Tactile graphics: an overview and resource guide. *Information Technology and Disabilities* 3, 4.

[13] R. I. Hartley and A. Zisserman. 2004. *Multiple View Geometry in Computer Vision* (second ed.). Cambridge University Press, ISBN: 0521540518.

[14] Rabia Jafri, Syed Abid Ali, and Hamid R Arabnia. 2013. Computer vision-based object recognition for the visually impaired using visual tags. In *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*. WorldComp), 1.

[15] Chandrika Jayant, Hanjie Ji, Samuel White, and Jeffrey P. Bigham. 2011. Supporting Blind Photography. In *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '11)*. ACM, 203–210.

[16] Sang-Hack Jung and Camillo J Taylor. 2001. Camera trajectory estimation using inertial sensor measurements and structure from motion results. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, Vol. 2. IEEE.

[17] 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 (CHI '17)*. ACM, New York, NY, USA, 5839–5849.

[18] Andrej Karpathy, Stephen Miller, and Li Fei-Fei. 2013. Object Discovery in 3D Scenes via Shape Analysis. In *International Conference on Robotics and Automation (ICRA)*.

[19] Harmanpreet Kaur, Mitchell Gordon, Yiwei Yang, Jeffrey P Bigham, Jaime Teevan, Ece Kamar, and Walter S Lasecki. 2017. Crowdmask: Using crowds to preserve privacy in crowd-powered systems via progressive filtering. In *Fifth AAAI Conference on Human Computation and Crowdsourcing*.

[20] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[21] Philip Kortum and Megan Johnson. 2013. The relationship between levels of user experience with a product and perceived system usability. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 57. SAGE Publications Sage CA: Los Angeles, CA, 197–201.

[22] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553, 436.

[23] Kyungjun Lee, Jonggi Hong, Simone Pimento, Ebrima Jarjue, and Hernisa Kacorri. Revisiting Blind Photography in the Context of Teachable Object Recognizers. In *Proceedings of the 16th International ACM SIGACCESS Conference on Computers & Accessibility*.

[24] Seeing AI. Microsoft. Accessed: 2019-12-29. https://www.microsoft.com/en-us/seeing-ai

[25] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. 2015. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics* 31, 5, 1147–1163.

[26] National Federation of the Blind. 2009. The braille literacy crisis in America: Facing the truth, reversing the trend, empowering the blind. http://www.nfb.org/images/nfb/documents/word/The_Braille_Literacy_Crisis_In_America.doc

[27] Jeffrey M Rzeszotarski and Meredith Ringel Morris. 2014. Estimating the social costs of friendsourcing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2735–2744.

[28] Miguel A Sánchez, Montserrat Mateos, Juan A Fraile, and David Pizarro. 2012. Touch Me: a new and easier way for accessibility using Smartphones and NFC. In *Highlights on Practical Applications of Agents and Multi-Agent Systems*. Springer, 307–314.

[29] Jeff Sauro. 2011. Measuring usability with the system usability scale (SUS).

[30] Jeremi Sudol, Orang Dialameh, Chuck Blanchard, and Tim Dorcey. 2010. Looktel-A comprehensive platform for computer-aided visual assistance. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. IEEE, 73–80.

[31] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2818–2826.

[32] TapTapSee. Accessed: 2019-12-29. http://www.taptapseeapp.com

[33] Tombari F. Navab N. Tateno, K. 2015. Real-Time and Scalable Incremental Segmentation on Dense SLAM.

[34] Ender Tekin and James M Coughlan. 2010. A mobile phone application enabling visually impaired users to find and read product barcodes. In *International Conference on Computers for Handicapped Persons*. Springer, 290–295.

[35] Amazon Mechanical Turk. Accessed: 2019-12-29. https://www.mturk.com/

[36] Marynel Vázquez and Aaron Steinfeld. 2014. An Assisted Photography Framework to Help Visually Impaired Users Properly Aim a Camera. *ACM Trans. Comput.-Hum. Interact.* 21, 5, Article 25, 29 pages.

[37] Elizabeth M. Wenzel, Scott S. Fisher, Philip K. Stone, and Scott H. Foster. 1990. A System for Three-dimensional Acoustic "Visualization" in a Virtual Environment Workstation. In *Proceedings of the 1st Conference on Visualization '90 (VIS '90)*. IEEE Computer Society Press, Los Alamitos, CA, USA, 329–337.

[38] Samuel White, Hanjie Ji, and Jeffrey P. Bigham. 2010. EasySnap: Real-time Audio Feedback for Blind Photography. In *Adjunct Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, 409–410.

[39] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. 2017. Random Erasing Data Augmentation. *arXiv preprint arXiv:1708.04896*.