ABSTRACT

Title of Dissertation: EXPLORING BLIND AND SIGHTED USERS' INTERACTIONS WITH ERROR-PRONE SPEECH AND IMAGE RECOGNITION

Jonggi Hong
Doctor of Philosophy, 2021

Dissertation Directed by: Assistant Professor Hernisa Kacorri
College of Information Studies

Speech and image recognition, already employed in many mainstream and assistive applications, hold great promise for increasing independence and improving the quality of life for people with visual impairments. However, their error-prone nature combined with challenges in visually inspecting errors can hold back their use for more independent living. This thesis explores blind users' challenges and strategies in handling speech and image recognition errors through non-visual interactions looking at both perspectives: that of an end-user interacting with already trained and deployed models such as automatic speech recognizer and image recognizers but also that of an end-user who is empowered to attune the model to their idiosyncratic characteristics such as teachable image recognizers. To better contextualize the findings and account for human factors beyond visual impairments, user studies also involve sighted participants on a parallel thread.

More specifically, Part I of this thesis explores blind and sighted participants' experience with speech recognition errors through audio-only interactions. Here, the recognition result from a pre-trained model is not being displayed; instead, it is played back through

text-to-speech. Through carefully engineered speech dictation tasks in both crowdsourcing and controlled-lab settings, this part investigates the percentage and type of errors that users miss, their strategies in identifying errors, as well as potential manipulations of the synthesized speech that may help users better identify the errors.

Part II investigates blind and sighted participants' experience with image recognition errors. Here, we consider both pre-trained image recognition models and those fine-tuned by the users. Through carefully engineered questions and tasks in both crowdsourcing and semi-controlled remote lab settings, this part investigates the percentage and type of errors that users miss, their strategies in identifying errors, as well as potential interfaces for accessing training examples that may help users better avoid prediction errors when fine-tuning models for personalization.

EXPLORING BLIND AND SIGHTED USERS' INTERACTIONS WITH ERROR-PRONE SPEECH AND IMAGE RECOGNITION

by

Jonggi Hong

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2021

Advisory Committee:
Assistant Professor Hernisa Kacorri, Chair/Advisor
Assistant Professor Marine Carpuat
Assistant Professor Huaishu Peng
Assistant Professor Zhicheng Liu
Associate Professor Leah Findlater (University of Washington)

Acknowledgments

First and foremost, I would like to thank my advisor, Professor Hernisa Kacorri, for her endless support and thoughtful advice throughout the years. She has been supportive in completing my research as well as becoming an independent researcher. Her passion and creative ideas inspired me many times. I believe that what I learned from her would be a good guideline as a researcher for the rest of my life.

I also would like to thank my former advisor, Professor Leah Findlater, who shaped my research at the beginning of my Ph.D. at UMD. She provided many important skills and techniques related to all parts of conducting research and building relationship with other researchers. I was very fortunate to have opportunities to work with Professor Hernisa Kaccori and Professor Leah Findlater.

I would like to thank my dissertation committee members: Marine Carpuat, Huaishu Peng, and Leo Zhicheng Liu. Thank you for providing insightful comments that made my work stronger.

I am grateful to all lab mates, students, and friends who have shared skills and ideas with me: Uran, Kotaro, Lee, Meethu, Kristin, Kyungjun, Utkarsh, Rie, Alisha, Christine, Ebrima, Jaina, Ernest, June, Tak, Deokgun, Soekbin, Seongkook, and Jaeyeon.

As always, I thank my parents' for providing a huge support in completing my Ph.D. successfully. I am also grateful to my sister for giving me lots of assistance.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Using deep neural networks and large datasets (*e.g.*, ImageNet [5], LibriSpeech [6]), recent machine learning systems have reduced errors dramatically. For example, the word error rate of the state-of-the-art speech recognition system is only around 5% for English [7]; top-5 error rate for image classification is roughly at 4% [8]. With advances in computer vision, speech recognition, and natural language processing, machine learning has been employed in a variety of applications such as self-driving cars, automated retail services (*e.g.*, Amazon Go), and voice-controlled intelligent personal assistants (*e.g.*, Google Home, Amazon Echo).

While we've reached low error rates in object and speech recognition tasks with public benchmark datasets (*e.g.*, lower than 5% error rates in speech recognition with the World Street Journal dataset [9] and top-5 image classification with ImageNet [5]), these error rates may not be reflective of real world scenarios. In practice, one would expect to see much higher numbers due to many factors such as difficult tasks (*e.g.*, the error rate of top-1 image classification is higher than top-5 classification as shown in Figure 1.1), limited computational resources (*e.g.*, classifying images locally on a mobile device), or inputs that deviates from the training data (*e.g.*, classifying images with personal items or unique backgrounds collected by a user). Such errors are known to affect the

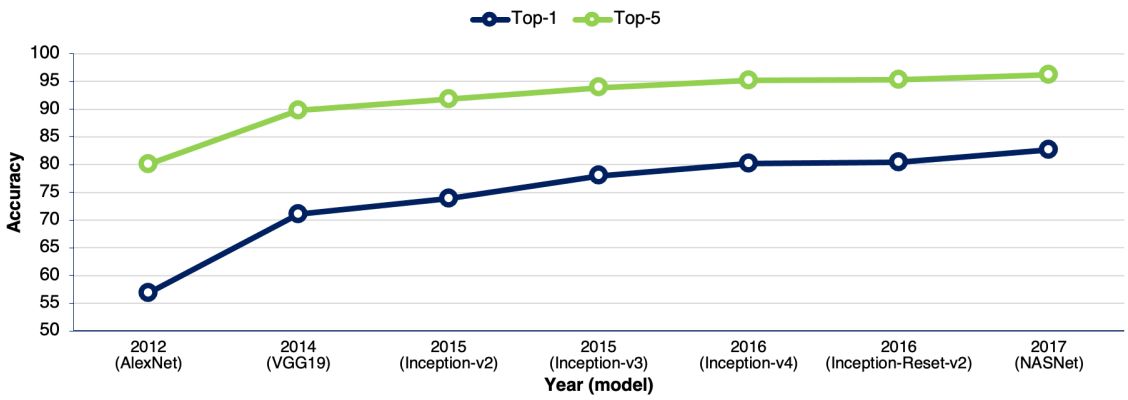

Figure 1.1: Upward trend and a plateauing line for top-1 and 5 accuracies for image classification on ImageNet from 2012 to 2017. (Source: Su *et al.*, 2018 [3])

user experience significantly in machine learning applications [10, 11]. For example, Fox *et al.* [12] identified causes of errors in automatic speech recognizer (ASR) such as similar sounding words, software parsing error, faulty microphones, and hardware issues. Reviewing and editing text to correct ASR errors has been known as a bottleneck in text entry through speech [13, 14, 15, 16]. In object recognition systems, errors can sneak in due to a mismatch of the training and real-world data [17] but also that these systems are sensitive to adversarial attacks [18, 19]. While the image classifiers are particularly useful for blind users [20], they would not be able to tell most of these errors especially if they could not touch the object recognized in the image (*e.g.*, recognizing a far object, a scene recognition). Therefore, interfaces that support users in identifying, correcting, understanding and altogether avoiding errors is crucial. In the broader HCI area, we see recent efforts such as Amershi *et al.* [10] providing guidelines of user interface design for AI-infused systems[1]. They emphasize that an interface ought to inform users how accurately the system can do its task and how to deal with the errors. As it is hard for users to predict and understand the errors from the AI-infused systems, researchers

[1] A term coined by Amershi *et al.*(2019) [10]

have put effort into explaining the model and its output to help users understand the rationale behind the system's output and the impact of a user's input on the behavior of the model [21]. This thesis is complementary to these efforts with a focus on everyday AI-infused systems employing speech and image recognition that can benefit the blind community and whose errors are typically identified through sight making them inaccessible to blind users.

Chapter 2: Background on User Interaction With Error-Prone Systems

This chapter provides an overview of user-error interaction in broader automation as well as machine learning systems with a focus on blind users' interactions with errors. As machine learning has been employed in many assistive technologies such as brain-computer interface [22, 23], sign language synthesis [24], and indoor localization [25, 26], developing accessible interfaces for identifying, understanding, and recovering from errors is essential. This chapter organizes prior studies based on the following steps to resolve errors: identifying, understanding, and recovering from errors.

2.1 Identifying Errors

Error identification plays an important role in the user-error interaction as a first step to handling errors. While errors are obvious and easy to tell in some applications where users can understand the outcome from the system and ground truth easily and quickly (*e.g.*, navigating familiar routes with way-finding system), the outcome from the system may not be clearly perceived due to the characteristics of the task, poorly designed interface, the complexity of the information, or poor concentration caused by a high workload, etc [27, 28]. For example, the ground truth may not be available immediately when the outcome is provided by the system (*e.g.*, medical diagnosis, weather prediction).

The ground truth may not be straightforward to the user if the system handles data in an unfamiliar work domain [29]. Therefore, researchers have explored the challenge of identifying machine learning errors. For example, the interface for identifying errors has been investigated because of the significant impact of the error handling process in speech, handwriting, and gesture recognition systems on the effectiveness of natural input [14, 15, 30, 31]. To resolve the challenges, prior studies have developed interfaces for helping users identify the errors. Bourguet [30] identified two approaches when categorizing these prior studies: *automatic identification*, where the system automatically identifies potential errors in its output and *machine-led discovery*, where the system has an interface that aids users to discover errors. However, while these approaches could reduce the errors missed by sighted users, it is still hard for blind users to identify all errors for several reasons. If the automatic identification systems missed any of the errors, blind users may not have a way to tell. Also, many systems with the machine-led discovery approach are designed for sighted users using visual feedback [30]. For example, a handwriting recognition system displayed different types of lines (*e.g.*, bold and dotted) to represent the system's top guess and the potential alternatives [32]. A common way to present alternative predictions for potential errors in speech recognition systems is by employing a graphical user interface to show *n-best* predictions. However, these approaches are not effective to enable blind users to identify errors easily because most of their interfaces depend on visual information which is not available for blind users. In consequence, prior studies on speech and image recognition systems revealed difficulties in identifying errors. While speech input is blind users' main method to enter texts on a mobile device, they have difficulties in identifying speech recognition errors due to the similarity between the

original speech input and the misrecognized texts in synthesized speech [13]. MacLeod *et al.* [33] conducted a user study to understand blind users' experience with social media images and showed that blind users were usually not able to identify the problems in the computer-generated captions of images even when the captions were incorrect and out of context. When errors occur in computer-generated captions, blind users tended to justify the difference between the caption and the text around the image rather than considering the captions as errors. These studies emphasize the importance of developing accessible interfaces for identifying and understanding errors.

2.2 Understanding Errors

Though deep neural networks and large datasets made a breakthrough in the performance of machine learning systems, the complexity of the systems made it difficult for users to understand the system's behavior. Therefore, prior studies have presented guidelines for interface design of machine learning applications to help users understand the output from the applications. For example, the guidelines developed by Amershi *et al.* [10] recommend informing users of what and how well a system can do to control users' expectations. Another prior study confirmed that controlling a user's expectation with the information of the system's capability impacts the user's experience positively when the system makes some errors [34]. As the complexity of the machine learning models makes it harder for users to understand why the errors occur, researchers have actively investigated the methods to explain the machine learning models and their output (*i.e.*, explainable artificial intelligence) [35, 36, 37, 38, 39, 40]. The explanations are found

to make users satisfied, perceive better control of the system, and trust the output of the system better [41, 42]. Therefore, many machine learning applications such as image classifiers [43], recommendation systems [35, 44], and news feed algorithms in social media [38] employed explainable interfaces.

As blind users depend on the machine learning-based assistive tools without visual information for some tasks (*e.g.*, writing texts with speech recognition or handwriting recognition [30, 45], finding routes with navigation system [46]), they care about understanding the errors and the consequence with them. Prior studies indeed showed that the errors impact blind users' experience with the systems significantly. For example, when it comes to self-driving vehicles for blind users, the safety issues caused by malfunctions in autopilot system is the main concern when they were suggested to use the self-driving vehicles independently without help from sighted people [47, 48]. Similar concerns exist with the systems where the risk of having errors is not as much as those with self-driving vehicles where errors may threaten users' lives. Though some errors in a navigation system are acceptable to blind users, they had a negative experience with errors when people around them showed misguided responses [49]. Another prior study showed that, when the navigation system guided a blind user to a place that is only a few meters away from the destination, blind users may be frustrated and get totally lost due to the small error [46]. It shows that the interface for user-error interaction needs to provide context or supplementary information with a prediction from a machine learning model so that blind users can figure out the cause and severity of errors accurately.

2.3 Avoiding and Correcting Errors

Prior studies have shown that the interactions for correcting errors have a significant impact on users' experience with machine learning systems [14, 15, 30, 31]. To improve the interactions with errors, several guidelines for designing user interfaces in machine learning applications recommend employing an intermediate step for users to provide a confirmation for the predictions from machine learning models [50, 51]. One of the most common approaches in recognition systems to allow users to correct errors is to repeat the input that was misrecognized by the system [30]. For example, in the case of handwriting recognition, a user overwrites the incorrectly recognized words for correction [52]. However, repeating has shown to be an inefficient way to correct the errors due to the possibility of having the same errors repeatedly [53, 54]. Therefore, to avoid the repeated misrecognitions with inputs in the same modality, a prior study recommended having multi-modal interactions for users to correct errors with different modalities [55]. For example, if speech input fails repeatedly, users can type the misrecognized text with a keyboard. A prior study confirmed the effectiveness of multi-modal interactions, showing that experienced users tend to switch modalities more often than the first-time users of speech recognition systems to correct errors [14].

In assistive tools for blind people, the easiness of correcting errors is a critical factor that affects the users' decision on whether they will continue to use the tools or not. Prior studies showed that people with disabilities frequently decide to use or abandon accessibility tools based on whether they can easily recover from errors or not [56, 57]. When it is difficult to recover from failures of some systems, users of

assistive technologies usually try to find mitigation strategies such as using multiple devices/software for the same task [58]. However, when multiple devices or applications are not available, blind users have to depend on their intuition or experience to recover from errors. For example, when a blind navigation system provides incorrect directional guidance, blind users find the correct directions based on their experience and awareness of the situation [26]. On the other hand, a prior study also showed that errors of only a few meters in a navigation system would make blind users frustrated and lost when the place is not familiar to them [46]. Therefore, given that many sources of errors exist in speech and image recognition systems, assistive tools based on these recognition systems need to provide user interfaces for reviewing, understanding, and recovering from errors.

Part I: Interacting with Error-Prone Speech Recognition

Prologue to Part I

Using deep neural networks, researchers have achieved vast improvements in speech recognition. Speech input is faster and more accurate on mobile devices than entering text with a touchscreen keyboard [59]. It is a primary means of text input on devices that have a small or no visual display, such as smartwatches or voice-based intelligent personal assistants (*e.g.*, Google Home, Amazon Echo). Speech input is also particularly useful for eyes-free interaction (*e.g.*, using a mobile device while walking or driving) or as accessible input for blind users [13, 60].

Reviewing and editing the inputted text, however, is a bottleneck [15]. Speech recognition errors arise from several sources: the ambiguity of words (*e.g.*, homophones and pronouns), background noise, and mistakes from users [61]. Visual interfaces for error detection and correction have been proposed and studied for desktops and mobile devices (*e.g.*, [62, 63, 64]). When visual output is available, users can read the recognized text and easily identify these errors. However, error identification is challenging when users can only hear an audio synthesis of the same text [13]. Azenkot *et al.* [13] showed that while speech is a primary text input method for blind users on a mobile device, 80% of the time is spent reviewing and correcting errors with synthesized audio of recognized texts.

Part I characterizes the challenges in identifying ASR errors through audio-only interactions. Since experiences and listening rates for synthesized speech differ across blind and sighted users due to differences in the experience with a screen reader, we analyzed the ability to identify ASR errors with blind and sighted users separately. Therefore, the first four user studies involve sighted participants recruited through Amazon Mechanical Turk not using a screen reader. Findings and insights from these initial studies are used to guide the experimental design for a follow-up in-depth study that investigates experiences and the ability to identify errors across blind and sighted participants in a lab setup. In all studies, error-identification accuracy is measured through speech dictation tasks.

Specifically, Part I of this thesis will explore each of the following research questions:

- **RQ1**: How frequently are ASR errors missed? (We will investigate RQ1 in Chapter 4)
- **RQ2**: Do different synthetic speech manipulations affect the user's accuracy of identifying ASR errors? (We will investigate RQ2 in Chapter 4)
- **RQ3**: For what tasks do blind and sighted users use ASR? (We will investigate RQ3 in Chapter 5)
- **RQ4**: How different are the experiences with speech dictation and listening between blind and sighted users? (We will investigate RQ4 in Chapter 5)
- **RQ5**: Is the accuracy of identifying ASR errors different between blind and sighted users? (We will investigate RQ5 in Chapter 5)
- **RQ6**: What are the blind and sighted users' strategies of pointing to ASR errors? (We will investigate RQ6 in Chapter 5)

Chapter 3: Background

ASR and synthesized speech have been employed in a variety of accessibility scenarios. Here, this chapter reviews state-of-the-art ASR systems, applications to accessibility with a focus on blind users, studies related to the comprehension of synthesized speech and ASR error through audio.

3.1 Automatic Speech Recognition and Error Identification

The performance of ASR systems have been improved with various techniques. The techniques can be categorized into three approaches: acoustic-phonetic approach, pattern recognition approach, and artificial intelligence approach [65, 66]. The early ASR systems were built with the acoustic-phonetic approach [67]. This approach have been particularly useful for various applications using speech sound such as multilingual speech recognition, accent classification, speech activity detection systems, etc [68]. The pattern recognition approach had been a dominant method to build an ASR system for decades before the artificial intelligence approach emerged with the advance of deep learning technique. The state-of-the art ASR systems employed artificial intelligence approach using a deep neural network, reaching 5% word error rate (WER)[1] recently [69].

[1] $WER = \frac{S+D+I}{N}$ where S is the number of substituted words, D is the number of deleted words, I is the number of inserted words, N is the number of all words in the reference.

Though the prior studies achieved quite low error rates in restricted environments (*e.g.*, noise-free sound, limited vocabulary, articulate speech), many factors such as speaker variation and noise may cause ASR errors in practice [61, 70]. Therefore, researchers have been exploring techniques for automatically detecting ASR errors to supplement ASR systems, which are inherently error-prone [71].

Prior work has attempted to detect ASR errors automatically or to help users identify ASR errors. A simple approach is to visually highlight words that are grammatically incorrect, which is common in mainstream mobile devices, or words that have low ASR confidence [72]. Researchers have also attempted to automatically detect ASR errors for enhancing speech-based interfaces (*e.g.*, confirming a voice request for clarification when the system detects a potential recognition error [73]). While recent studies have developed methods to predict ASR errors using neural networks [73, 74, 75], the predictions reach 70% precision and 60% recall at best, suggesting that this is an open area of research.

The focus of our study is on identifying ASR errors by users in non-visual context, but a follow-on step is to correct those errors by editing the dictated text. With the exception of Azenkot *et al.* [13], already discussed, work on editing ASR results has assumed that users will visually review and edit the text. These visual editing approaches can be defined as unimodal (speech used edit) and multi-modal (other input modalities used to edit) [31]. Multimodal solutions have combined speech with modalities such as pen, touchscreen, and keyboard input [76, 77, 78, 79]. As an example of unimodal (speech only) correction, Choi *et al.* [80] developed a prediction model for distinguishing whether a user's utterance is intended to be a dictation input or a correction command, achieving 84% accuracy in offline experiments. However, unimodal interfaces suffer from

cascading side effects where speech input commands for correcting errors cause further ASR errors [15].

3.2 Automatic Speech Recognition for Accessibility

People with disabilities have been early adopters of user interfaces with speech input. Speech input can allow for efficient control of a computer, home-based IPAs (*e.g.*, Amazon Echo, Google Home), or mobile device for people with visual (*e.g.*, [81, 82, 83, 84]) or motor impairments (*e.g.*, [85, 86, 87, 88, 89]). ASR can also provide access to spoken information for people who are Deaf/deaf or hard-of-hearing (*e.g.*, [90]). For people with speech impairments, speech input has been used to support self-assessment of pronunciation (*e.g.*, [91, 92]) and to recognize a user's dictation and reproduce it through a synthesized voice (*e.g.*, [93]).

In the Chapter 4 and 5, we characterize the strategies and challenges in detecting ASR errors using synthesized speech (*i.e.*, text-to-speech) among blind and sighted users. When comparing these two user groups, prior work has shown that blind users make use of speech dictation on mobile devices more often than sighted users [94], likely due to the inefficiency of using touchscreen keyboards with a screen reader [60]. Blind users also make use of speech input to access smartphone apps [84] and to browse the web [81, 82]. While in the latter cases, users can infer errors based on system response (*e.g.*, which app opens), for dictation tasks, ASR errors need to be identified by listening to the text-to-speech output from the screen reader. This ASR identification task is the focus of our study.

3.3 Comprehension of Synthesized Speech

Several studies have concluded that blind people comprehend synthesized speech better than sighted people. For example, Papadopoulos and Koustriava [95] showed that the comprehensibility of synthesized speech was higher for blind users, probably due to greater experience with screen readers, while natural speech was easier to understand than synthesized speech for both blind and sighted users. Similarly, Stent *et al.* [96] showed that users' experience with synthesized speech positively impacts the accuracy of transcribing fast synthesized speech; they tested 300 to 500 words per minute (WPM) speech rates with users with early-onset blindness. A recent study by Bragg *et al.* [97] measured the accuracy of answering questions based on synthesized speech ranging from 100 to 800 WPM, and found that the maximum intelligible speech rate was higher for blind users than sighted users. Blind users have also rated the degree of understanding ultra-fast synthesized speech, at a rate of 17-22 syllables per second (680-880 WPM), higher than sighted users [98]. However, the differences between these two groups of users may disappear when there are multiple streams of speech, called the Cocktail Party environment [99]. In support of this, Guerreiro and Gonçalves [100] found no differences between blind and sighted users in being able to focus on a specific source when exposed to 2-4 synthesized concurrent speech sources.

While the above studies evaluated the intelligibility and comprehensibility of synthesized speech and compared performance for blind and sighted people, they focused on speech output without errors, which contrasts our focus on identifying ASR errors through synthesized speech.

Chapter 4: Characterizing the Challenges in Identifying ASR Errors With Sighted Users

4.1 Motivation and Introduction

In this chapter, we quantify the problem of identifying speech recognition errors through audio-only feedback and investigate potential solutions. While researchers have examined understanding of and ability to transcribe synthesized speech output (*e.g.*, [96, 98]), the impacts of different synthesized speech manipulations on the user's ability to identify speech recognition errors have not been investigated.

We report on a series of four controlled studies. The goal of the first study was to characterize the problem of identifying errors based on audio-only output. For this in-lab study, native and non-native English speakers dictated and listened to the recognized version of a series of phrases in silent and noisy conditions. Overall, participants were unable to identify more than 50% of recognition errors when listening to the audio of the recognized text, with the most common difficulty being with multiple-word errors (*e.g.*, "mean" to "me in", or "storm redoubles" to "stormy doubles"). Studies in Chapter 4.2 through 4.5 then investigated the effect of three synthesized speech manipulations (*i.e.*, pauses between words, speech rate, and speech repetition) on the user's ability to identify

those recognition errors. Inserting pauses, in particular, could help to address the multiple-word errors identified from the study in Chapter 4.2. Studies in Chapter 4.3 and 4.4 showed that adding a pause between words resulted in significantly higher error identification rates than no pause, and that fast speech (*i.e.*, 300 WPM) made identification more difficult. Finally, The study in Chapter 4.5 evaluated another alternative—repeating the audio output twice—and found that repetition did not improve participants' ability to identify errors over simply listening to the audio once.

4.2 Understanding Error Identification in Recognized Speech

Though previous studies have shown that reviewing dictated text using non-visual output is a challenge [13], the extent of that challenge and the specific difficulties that users encounter have not been quantitatively assessed. How many misrecognized words do users miss when reviewing only through audio? What kind of errors is the hardest for users to identify? To answer these questions, we conducted a lab-based study where participants dictated a set of phrases using a mobile device and reviewed the system's recognition of each phrase by listening to audio output. To increase the generalizability of the findings, we manipulated the level of background noise and participants' fluency levels, two factors that are known to impact speech recognition accuracy [101].

4.2.1 Method

This controlled experiment measured the impacts of background noise level and the user's English proficiency, on the WER of the speech recognizer and on the user's ability

Figure 4.1: The experimental setup.

to identify recognition errors based on synthesized speech output.

Participants. We recruited 12 native English speakers (5 male and 7 female) and 12 non-native English speakers (8 male and 4 female) through campus email lists. The native English speakers ranged in age from 18 to 36 (M=23.4, SD=5.3), while the non-native English speakers were 22 to 38 years old (M=26.3, SD=4.4). None reported having hearing loss. Non-native speakers had lived in the United States for 0.3 years on average (SD=2.3). Ten native speakers and eight non-native speakers had experience with speech input before, while the remaining participants did not.

Procedure. Study sessions took 30 minutes and were conducted in a quiet room. As shown in Figure 4.1, participants sat at a table on which a Galaxy Tab 4 and two speakers were placed. We first collected demographic information and experience using speech input. The silent and noisy conditions were then presented in counterbalanced order. The tablet's audio output was set to 75% of maximum volume, which was approximately 60db with the synthesized speech audio. For the noisy condition, the speakers played

street noise at 50db. A custom Android application guided participants through 30 trials per condition, where each trial consisted of: (1) reading a phrase displayed on the tablet screen, (2) dictating the phrase, which included double-tapping the screen to indicate the start and end of dictation, (3) listening to synthesized speech output of the recognized phrase, and (4) identifying discrepancies, if any, between the dictated and recognized text. This lattermost step involved reporting words that had been incorrectly recognized and locations where extra words were inserted. Participants viewed the reference phrase while listening to the synthesized speech, and verbally reported errors they heard to the experimenter.

The phrases were randomly selected without replacement from a set with 200 phrases extracted from the LibriSpeech ASR corpus [6]. Of the 2703 phrases in the Librispeech development subset, 600 had 10 or fewer words, of which we randomly selected 200 that were of a complete sentence form, comprehensible, and contained no proper nouns which would increase ASR errors. The IBM Speech-to-Text API was used for speech recognition because it provides functions to analyze the speech recognition results (*e.g.*, confidence scores and timing of words). The speech was synthesized on the tablet device using the TextToSpeech function in Android 5.0 with the default speech rate of 175 WPM (which is within the range recommended in the research literature as well [102]).

4.2.1.1 Data Analysis

Study Design. This study used a mixed factorial design with a within-subjects factor of Noise (silent vs. noisy) and a between-subjects factor of Fluency (native vs.

non-native). The silent and noisy conditions were presented in counterbalanced order. Participants were randomly assigned to orders.

Measures and Data Analysis. To provide a baseline understanding of how well the speech recognizer performed, we computed word error rate (WER) on the recognized text [72]; lower rates are better. To assess the user's ability to identify errors, we computed precision—that is, when a participant thinks they hear an error, how often is it actually an error—and recall—that is, how the proportion of true errors participants were able to identify. We also employed phrase-level accuracy as a secondary measure, that is, whether a participant identified at least one error in a phrase that contains one or more errors, or no errors in a correct phrase. For this exploratory study, we focused on accuracy and did not measure speed.

To compute these measures, we needed to judge whether each instance where the participant pointed out an incorrectly recognized or inserted word was a true positive, or that the lack of an error label was a true negative. Ambiguity arose when a single word was recognized as multiple words (*e.g.*, "meet" to "me it"). Is this (i) one "incorrect word – meet" or (ii) one "incorrect word – meet" plus one "inserted word – it"? We considered both responses to be correct, with (i) counted as a truly positive and (ii) counted as two true positives. As a third case, if the participant marked this error as simply one "inserted word – it", we judged the response to include one false negative (the word "meet" should have been marked as incorrect) and one true positive (for the word "it" being added).

WER and precision violated the normality assumption of an ANOVA (Shapiro Wilk tests, $p < .05$), so we instead used 2-way repeated-measures ANOVAs with aligned rank transform (ART) for these measures, a non-parametric alternative to a factorial

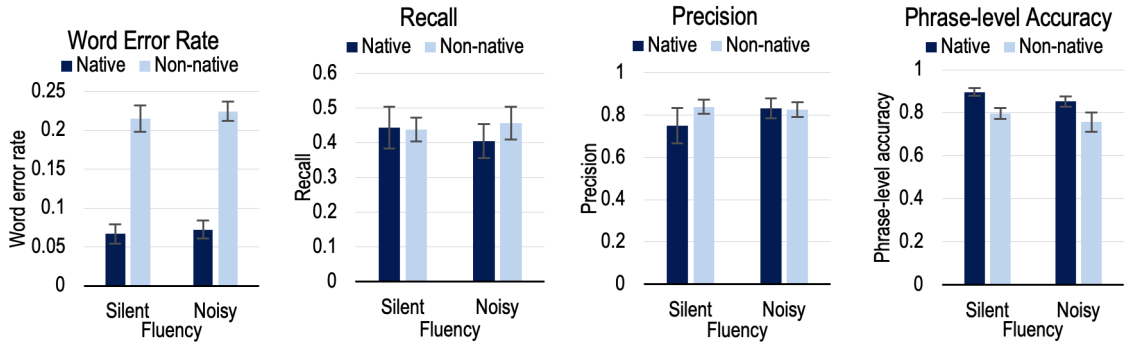

Figure 4.2: WER, precision, recall, and phrase-level accuracy in Study 1. Recall results showed that participants missed identifying more than half of the speech recognition errors. Error bars show standard error ($N = 12$ per group).

ANOVA [103]. The recall was analyzed in the same way, but without the ART adjustment.

4.2.2 Results

Figure 4.2 shows the WER, precision, recall, and phrase-level accuracy in silent and noisy conditions with native and non-native speakers.

Fluency affected speech recognition accuracy. Impacts of fluency and noise on WER have been previously studied, so our intention in including these factors in the experience was simply to increase the generalizability of our main measures (*i.e.*, precision and recall in identifying recognition errors). For completeness, however, we still examined whether fluency and noise impacted WER. As expected based on past work [104], fluency did impact WER. WER was higher with non-native speakers than native speakers, at 0.22 (SD=0.05) compared to 0.07 (SD=0.04); this difference was significant (main effect of Fluency: $F_{1,22} = 123.00, p < .001, \eta^2 = 0.74$). The WER with native speakers was close to typical WERs achieved by recent speech recognition engines at 0.05-0.10 WER [7]. Different levels of background noise did not significantly impact WER (main effect of

Noise: $F_{1,22} = 0.79, p = .384, \eta^2 < 0.01$), nor was there a significant interaction effect between Fluency and Noise ($F_{1,22} = 0.16, p = .693, \eta^2 < 0.01$).

Participants missed more than half of the errors. In terms of participants' ability to identify the speech recognition errors based on audio output, across all conditions, precision was 0.81 (SD=0.18), meaning that 19% of the errors that participants marked were not true errors. Of greater importance for being able to produce accurate text input, however, are the relatively low recall rates: on average across all four conditions, only 0.44 (SD=0.16) of true errors were identified—more than half the errors were undetected. Phrase-level accuracy, which could allow a user to at least know they should re-dictate an entire phrase even if they are not aware of all detailed errors, was higher, at 0.90 (SD=0.06) in the best case (native speakers + silent).

ANOVA (with ART if applicable) results revealed no significant main or interaction effects of Fluency or Noise on precision or recall. There was a significant main effect of Fluency on phrase-level accuracy ($F_{1,22} = 7.48, p = .009, \eta^2 = 0.14$), whereby native speakers had higher accuracy than non-native speakers, at 0.85 (SD=0.08) compared to 0.76 (SD=0.12). However, the main effect of Noise and the interaction effect between Fluency and Noise on phrase-level accuracy were not significant.

Multiple-word errors were most difficult to identify. To better understand what types of errors participants had trouble identifying, we qualitatively analyzed the 183 errors that native speaker participants missed (*i.e.*, instances of false negatives). Native speakers who are most likely to use speech input in English were target participants in Studies 2-4, so we focused on native speakers in this analysis. One research team member coded the missed errors into the categories below. For validation, a second coder also independently coded

all missed errors, and Cohen's kappa showed strong inter-rater agreement (kappa=0.82, 95% CI: [0.76, 0.88]). The categorizations were as follows:

- *Multiple-word errors (N=107; 58.5%).* Multiple sequential words sometimes sounded like another word or words. We included cases where multiple words were recognized as a single word (*e.g.*, 'a while' and 'awhile'), multiple words were recognized as other multiple words (*e.g.*, 'storm redoubles' and 'stormy doubles'), and single words were recognized as multiple words (*e.g.*, 'meet' and 'me it').

- *Single word errors (N=57; 31.1%).* This type of error includes single words that were replaced with homophones or other single words with similar sounds (*e.g.*'inquire' and 'acquire', 'he' and 'she').

- *Punctuation mark errors (N=7; 3.8%).* There is typically no explicit indication of punctuation marks such as apostrophes in text-to-speech output. If the recognized word is exactly the same as the intended word except for a punctuation mark, we classified it as a punctuation error (*e.g.*, 'state's' and 'states').

- *Other (N=12; 6.6%).* In some cases, the type of error was unclear. For example, when there were many errors in a phrase the participant may simply have been unable to remember them all.

4.2.3 Summary and Discussion

Across both user groups, participants missed over 50% of recognition errors when listening to the audio playback. Phrase-level accuracy, which would allow a participant

to know they should re-dictate an entire phrase, was higher but still left many unidentified errors (10% of phrases). The majority of the errors that participants did not notice were classified as multiple-word errors. A potential solution to address this type of error is to emphasize the individual words in the text-to-speech output by adding pauses between words—an approach that we focus on in Studies 2-4 alongside other simple output manipulations. That there were no differences in WER or participants' ability to identify recognition errors between different background noise levels suggests that we may have needed a wider range of noise levels to properly assess that factor.

4.3 Improving Error Identification Through Speech Rate and Pause

Study 1 showed that participants missed a substantial number of errors when listening to the confirmation audio clips, with the most common type of missed error being a multiple-word error. In Study 2, we focused on a straightforward potential means of addressing this problem: adding artificial pauses between words in the speech output, which should allow the user more easily distinguish individual words. Inserting pauses in synthesized speech affects prosody and elision—the latter being when successive words are strung together while speaking, causing the omission of an initial or final sound in a word. While this change is not ideal for many uses of text-to-speech, it is potentially useful for helping users to correct recognition errors with audio-only interaction.

This study isolated the error identification component of speech input and correction. Fifty-four crowdsourced read a series of phrases that had been dictated in Study 1, listened to corresponding confirmation audio clips (*i.e.*, text-to-speech output of what the system

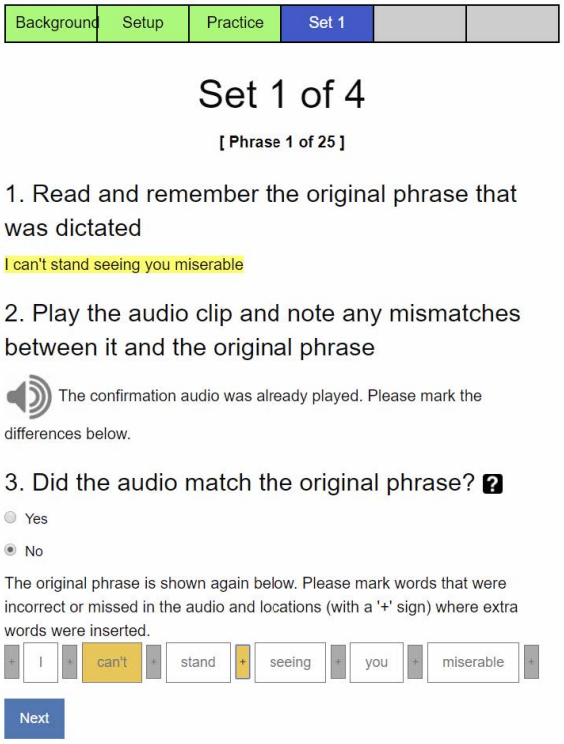

Figure 4.3: Screenshot of the online testbed used for Studies 2, 3, and 4, showing a single trial. A trial consisted of reading a presented phrase, listening to an audio clip of what a speech recognition engine had heard, and marking errors in the recognized version (*i.e.*, discrepancies between text and audio)..

had recognized), and identified discrepancies (recognition errors) between the presented text and the audio output under varying conditions: no/short/long pause and three speech rates.

4.3.1 Method

For this study and the two subsequent ones, we recruited crowdsourced participants on Amazon's Mechanical Turk to be able to run a series of studies with a larger and more diverse sample than would have been feasible in the lab.

Participants. The 54 participants (33 male, 21 female) ranged in age from 21 to 58 (M=33.4, SD=8.8). All participants were native English speakers, and none reported

hearing loss. Just over half (N=29) had experience in using speech input. All participants reported completing the study in a quiet room.

Procedure. Participants were directed to an online testbed that guided them through the 45-minute study procedure. The procedure began with a background questionnaire, followed by instructions about the overall tasks. Participants were then shown a sample phrase and asked to adjust the sound volume to ensure they could easily hear the audio clip.

The task consisted of identifying discrepancies between presented text phrases and audio clips, where the audio clips may contain errors made by a speech recognizer. To test realistic speech recognition errors, the pairings of presented phrases and audio clips were taken from the speech input collected during Study 1. That study resulted in 600 pairs of presented and recognized phrases, where 32.4% of the recognized phrases included at least one error. The Say app in Mac OS X was used to generate the synthesized speech, including pauses.

Figure 4.3 shows an example trial, with the presented phrase and an audio clip widget. After clicking to listen to the audio clip once (a single time; no replays allowed), the participant answered (yes/no) whether the audio clip had matched the presented phrase. The page included boxes that mapped to each word in the presented phrase as well as locations before and afterwords where extra words could appear. Participants marked all discrepancies between the presented text and the audio by clicking the corresponding boxes. The boxes were only enabled after the audio clip finished playing, so participants could not mark errors while actively listening to the audio. The presented phrase was visible for the duration of the trial. The 'next' button was enabled only after the participant

had reported whether the audio contained any errors.

Participants first completed six practice trials to familiarize themselves with the task. Practice trials used a typical text-to-speech output setting of 180 WPM and no pauses between words. After each practice trial, participants were shown the correct answer as feedback. The experimental conditions were then presented in counterbalanced order, with 20 test trials per condition. Phrases were randomly selected from the set of 600 with no replacement, and different phrases were used for practice and test trials. After finishing all conditions, participants had to answer questions about easiness and preference of conditions.

Study design. Study 2 used a 3x3 within-subjects design with factors of Speech Rate (100, 200, and 300 WPM) and Pause Length (no pause, 1ms, and 150ms). Order or presentation for the nine conditions was counterbalanced using a balanced Latin square (in fact, two squares due to having an odd number of conditions). Participants were randomly assigned to orders.

The 1ms pauses, while too short to cause a detectable silence in the output, were used to eliminate elision in contrast to the 'no pause' condition. The 150ms pause length was selected based on pilot testing different lengths (1 to 200ms) to identify a short, yet distinguishable pause. Because the effectiveness of pause lengths and error identification, in general, may be impacted by the speech rate, we included three speech rates: one close to default rates in commercial text-to-speech systems (200 WPM), a slower rate (100 WPM), and a faster rate (300 WPM).

4.3.1.1 Data Analysis

Like Study 1, we computed precision, recall, and phrase-level accuracy of identifying errors in the audio clips. Two participants were excluded from the analysis because they did not mark any words as errors in one condition, making it impossible to calculate the precision. Although our focus is on how well participants identify errors, for completeness we also report on trial completion time (time from the start of a trial to clicking the 'next' button).

However, low trial completion times are not necessarily our goal, since they could be due to not noticing and thus not taking the time to mark errors. Perhaps more importantly, the length of the audio clips varies by condition, so we also report on descriptive statistics for audio clip length.

Precision, phrase-level accuracy, and trial completion time violated the normality assumption of ANOVA (Shapiro-Wilk tests, $p < .05$). Therefore, 2-way repeated-measures ANOVAs with ART was used, with Wilcoxon signed-rank tests and a Bonferroni correction for posthoc pairwise comparisons. For recall, a 2-way RM ANOVA was used with paired t-tests for posthoc pairwise comparisons.

4.3.2 Results

Figure 4.4 shows our primary measures of precision, recall, and phrase-level accuracy, along with trial completion time for completeness.

Pauses and slower speech improve recall. Recall ranged from 0.48 to 0.67 across the nine conditions. Pause Length significantly impacted recall ($F_{2,408} = 1.47, p <$

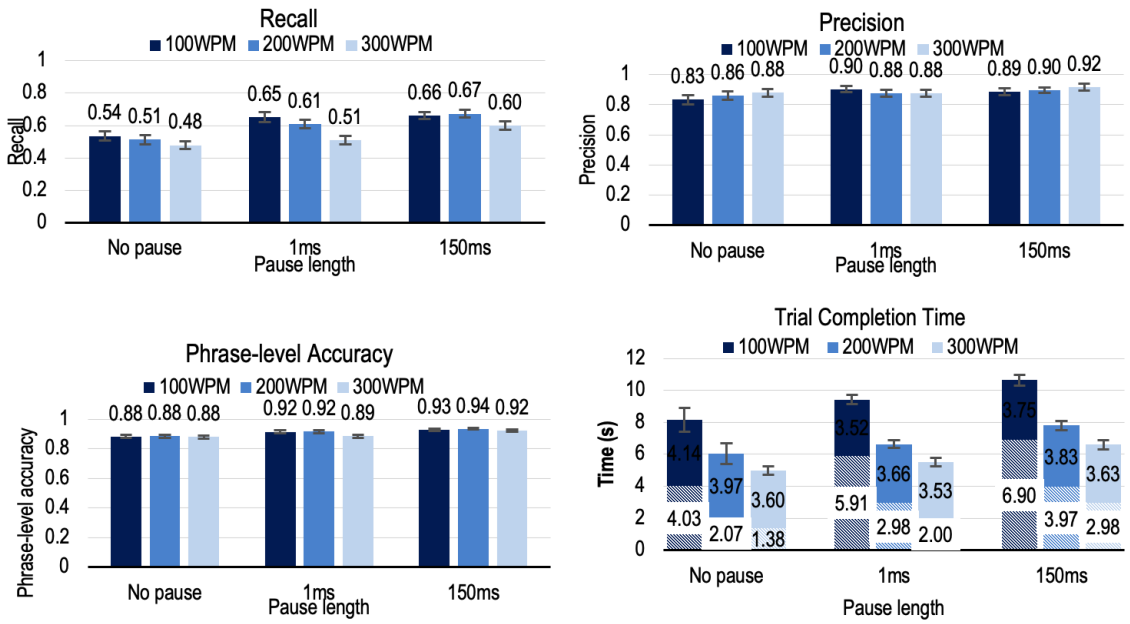

Figure 4.4: Precision, recall, phrase-level accuracy, and trial completion time in Study 2. The shaded portion in trial completion time indicates the average length of audio clips in that condition. Participants identified errors most accurately with the 200 WPM speech rate and 150ms pause. Error bars show the standard error ($N = 52$).

$.001, \eta^2 = 0.07$). All posthoc pairwise comparisons were significant ($p < .05$), showing that as pause length increased, so did recall. Speech Rate also significantly affected recall ($F_{2,408} = 0.66, p < .001, \eta^2 = 0.03$). Posthoc pairwise comparisons showed that the 300WPM speech rate resulted in significantly lower recall than the other two speeds (both comparisons $p < .05$). The interaction between Speech Rate and Pause Length was not significant ($_{F4,408} = 0.89, p = .467, \eta^2 < 0.01$).

Pauses also impact precision. Precision ranged from 0.83 to 0.92 across the nine conditions. Precision was significantly impacted by Pause Length ($F_{2,408} = 3.71, p = .025, \eta^2 = 0.01$), although after a Bonferroni correction no posthoc pairwise comparisons were significant. There was no significant main effect of Speech Rate on precision ($F_{2,408} = 1.22, p = .297, \eta^2 < 0.01$), nor was the Pause Length x Speech Rate interaction

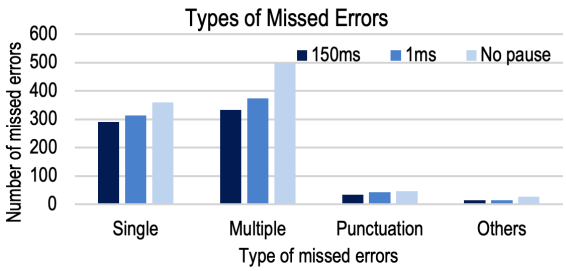

Figure 4.5: Types of errors participants missed identifying in Study 2. Participants missed 33% fewer multiple-word errors with a 150ms pause compared to no pause.

effect significant ($F_{2,408} = 1.53, p = .193, \eta^2 < 0.01$).

Secondarily, pauses improve phrase-level accuracy. Overall, Pause Length significantly impacted phrase-level accuracy ($F_{2,408} = 22.36, p < .001, \eta^2 = 0.07$), with posthoc pairwise comparisons, showed that the differences between all pairs of pause lengths were significant (all p¡.05). Speech Rate also significantly impacted phrase-level accuracy ($F_{2,408} = 6.46, p < .001, \eta^2 = 0.01$), but no posthoc pairwise comparisons were significant after a Bonferroni correction. The interaction effect between Speech Rate and Pause Length was not significant ($F_{4,408} = 2.31, p = .058, \eta^2 < 0.01$).

Trial completion times and audio lengths as expected. The length of time to play the audio clip consisted of a substantial portion of the trial completion time on average, as shown in Figure 4.4. The downside of inserting pauses between words and slowing down speech playback is that these changes lengthen the audio clip time. Accordingly, there was wide variation in both trial completion times and audio clip length. Even the 1ms pause added 10-15% to trial completion times across the three speech rates compared to no pause, and 44-47% if just examining the length of the audio clips because the pauses eliminate overlaps between words (eliminating elision).

Identifying multiple-word errors improved the most. To examine the effect of pauses

on specific types of errors, we manually coded 2341 missed errors from all participants. Figure 4.5 shows the number of errors of all types. The overall trend shows that all three types of errors decreased as the pause increased. However, the most substantial reduction was for multiple-word errors, which dropped 33.2% from the no pause condition (497 missed errors) to the 150ms pause condition (332 errors). In contrast, missed single-word errors only dropped 18.9%, from 359 to 291, and punctuation errors dropped 28.2%, from 46 to 33.

	Speech rate (WPM)			**Pause length (ms)**		
	100	200	300	no	1	150
Ease	21	27	4	17	21	14
Preference	7	31	14	20	22	10

Table 4.1: Subjective vote tallies in Study 2. The 200 WPM speech rate and shortest two pause lengths were the most preferred, while 300 WPM was least likely to be voted easiest.

Speech rate impacted perceived ease and preference. The subjective responses differed from the objective measures. Table 4.1 shows vote tallies for easiest and most preferred speech rates and pause lengths. Pearson Chi-Square test of independence showed that Speech Rate significantly impacted ease ($X^2_{(2,N=52)} = 16.42, p < .001$) and preference votes ($X^2_{(2,N=52)} = 17.58, p < .001$). The 200 WPM speech rate received the most votes for both ease and preference. Pause Length did not significantly impact either measure. In open-ended comments, participants said that 200 WPM felt natural because it was close to normal speech rate. While the accuracy with 150ms was highest, nine participants felt that it sounded unnatural compared to the other two pause lengths. Four participants reported that the 1ms pause, however, gave a moment to think as well as being more natural than the 150ms pause.

4.3.3 Summary and Discussion

Recall and phrase-level accuracy were highest with the longest pause length (150ms), while the fastest speech rate (300 WPM) negatively affected recall. An important consideration, however, is that inserting pauses and slowing down speech increases audio clip length and thus overall task time. Compared to the baseline condition (*i.e.*, 200 WPM, no pause), the best combination (200 WPM, 150ms pause) resulted in a 31% increase in recall and a 4% increase in precision, though also almost doubled the playback length. Even the 1ms pause made the audio 0.6-1.9s longer than no pause audio because it removed the elision in the phrase. In terms of subjective responses, most participants preferred the 200 WPM speech rate (which corresponds to [102]) and felt that 300 WPM made the task harder. However, there was no impact of pause length on subjective measures, suggesting that these short pauses (1ms, 150ms) may be acceptable compared to no pause even though they add time to the task.

4.4 Improving Error Identification by Varying Pause Length

Study 2 showed that inserting pauses between words in the speech output enables users to identify errors more accurately, but only included two pause lengths that were greater than 0ms. Because adding pauses increases overall task time, we would ideally be able to pinpoint the shortest pause length that is still effective, and use that during audio-only speech input. To more precisely identify an ideal pause length than was possible in Study 2, here we evaluate seven pause lengths ranging from 1ms to 300ms.

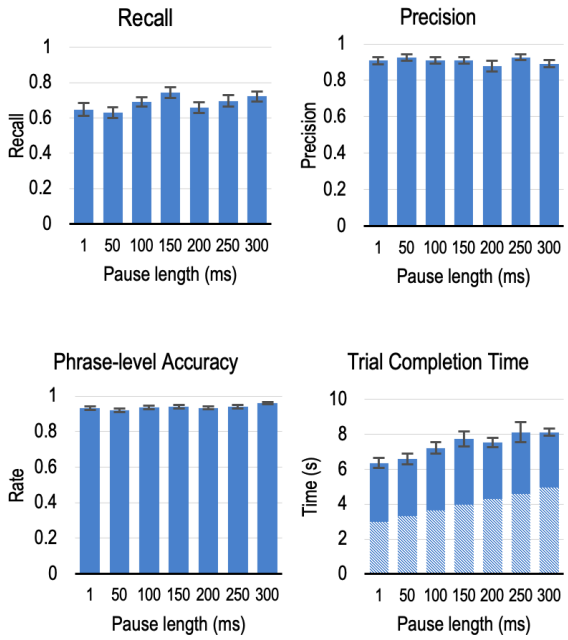

Figure 4.6: Graphs of precision, recall, phrase-level accuracy, and trial completion time in Study 3. The shaded portion in trial completion time indicates the length of audio clips. There were no significant differences in accuracy measures due to pause length. Error bars show standard error ($N = 40$).

4.4.1 Method

The study method is similar to Study 2 with the exceptions described here. The speech rate was fixed at 200 WPM because there were no significant error identification differences between 100 and 200 WPM in Study 2, but participants preferred 200 WPM. We recruited 42 participants (23 male, 19 female). Participants were on average 37.2 years old (SD=11.7; range 21-68). All were native English speakers and none had hearing loss. Twenty had previously used speech input. Four participants reported completing the study with light background noise (*e.g.*, light street noise or office), while the remaining 38 participants reported using a quiet room.

This study employed a within-subjects design with the single factor of Pause Length

(1, 50, 100, 150, 200, 250, or 300ms). This range spans from imperceptible pauses to highly obvious pauses. The seven conditions were presented in counterbalanced order using a balanced Latin square, similar to Study 2. Participants were randomly assigned to orders. Precision, recall, phrase-level accuracy, and trial completion time all violated the normality assumption of ANOVA (Shapiro-Wilk tests, $p < .05$), so 2-way RM ANOVAs with ART were used. Two participants who marked no errors in one condition were excluded from analysis because their precision could not be calculated.

4.4.2 Results

Figure 4.6 shows results for the four main measures. Unlike in Study 2, there were no significant main effects of Pause Length on recall, precision, or phrase-level accuracy (respectively: $F_{6,234=2.12}, p = .052, \eta^2 = 0.04; F_{6,234=0.68}, p = .667, \eta^2 = 0.02; F_{6,234=2.09}, p = .06, \eta^2 = 0.04$). Average audio clip length ranged from 3.0s per trial with the 1ms pause to 5.0s per trial with the 300ms pause. The trial completion time was shortest with 1ms pause at 6.4s and longest at 8.1s for both the 250ms and 300ms pauses. Following the performance results, there was no statistically significant difference in easiness and preference due to Pause Length (Chi-square tests, $p > .05$).

4.4.3 Summary and Discussion

These results are unexpected and appear to contradict Study 2, where we had concluded that the 150ms pauses resulted in significantly higher recall and phrase-level accuracy than the 1ms pause. (Note that the worst-performing condition from Study 2 – no pause

– is not included in this study.)

To confirm that the result of Study 3 was not obtained by chance and to better understand this unexpected result, we conducted two additional studies, which we report on briefly. First, we approximately replicated Study 3, but with 30 participants and two adjustments to increase statistical power: only four pause length conditions (1, 75, 150, and 225ms), and 40 trials per condition instead of 20. This replication yielded a similar result to what is reported above: no significant effects of pause length on recall, precision, and phrase-level accuracy. A subsequent closer examination of the Study 2 results, however, revealed that an important yet not statistically significant interaction effect may have affected those earlier conclusions: the 1ms vs. 150ms pause difference may have arisen primarily from the 300 WPM speech rate condition, rather than the 100 WPM or 200 WPM conditions. As such, because we used only 200 WPM in Study 3, we revisited the 200 WPM data from Study 2. A simple paired t-test showed that there was no significant difference between the 1ms and 150ms pause for recall; similarly, Wilcoxon signed-rank tests were not significant for precision or phrase-level accuracy. As such, Study 3 does confirm Study 2 but also provides more nuance on the conclusions.

Again, the worst-performing pause length from Study 2 was the *no pause* condition, which allowed us to conclude that inserting even 1ms pauses was better than no pause. To confirm that this conclusion still held for a 200 WPM speech rate alone, we first conducted a t-test and Wilcoxon signed-rank tests on the 200 WPM data from Study 2. The 1ms pause resulted in significantly higher recall and phrase-level accuracy than no pause (all $p < .05$). We then conducted a short follow-up replication: we collected new data from 28 participants who completed 25 trials in two conditions: 200 WPM with a 1ms pause

and 200 WPM with no pause. The 1ms pause resulted in significantly higher recall (t-test, $t_27 = 2.73, p = .011, d = 0.59$) and phrase-level accuracy (Wilcoxon signed-rank test, $W = 216.5, Z = 2.43, p = .014, r = 0.32$) than no pause.

Considering the results from both Study 2 and 3, we can conclude that inserting a pause between words does help significantly in identifying speech recognition errors at the preferred speech rate of 200 WPM, but the length of that pause does not matter. What is most important is the existence of a pause, perhaps because it eliminates elision.

4.5 Improving Error Identification by Listening to Speech Twice

Inserting pauses between words lengthens the time for audio playback. As already mentioned, even with only a 1ms pause, there was an additional 45% for playback time over no pause with the text-to-speech engine we used in Study 2. In this final study, we conducted an initial assessment of an alternative approach to making use of extra time: simply repeating the audio clip twice compared to listening to it only once. Participants in the earlier studies had only been allowed to listen to each audio clip once, to assess their first-pass ability to identify errors. However, repeating the audio twice could improve error identification. While it may be useful to assess the effects of repetition in more detail in future work, for this first evaluation, we compared clips at 200 WPM played only once versus played twice, with no pauses between words.

4.5.1 Method

The method is the same as for Study 2 except as follows . **Participants.** We recruited 30 participants (17 male, 13 female). Participants ranged in age from 23 to 66 (M=36.6, SD=11.5). All participants were native English speakers with no hearing loss. All participants reported completing the task in a quiet room, except for one who reported light background noise. Seventeen participants had previously used speech input on their phone or computer.

Study Design. We used a within-subjects design with two conditions: Default or Repeat. With Default, the audio feedback played once at 200 WPM with no pause between words, whereas with Repeat the audio played twice at 200 WPM with no pause between words and a chime sound (1.1s long) between repetitions. The two conditions were presented in counterbalanced order. Participants were randomly assigned to orders. Precision, recall, phrase-level accuracy, and trial completion time data all violated the normality assumption (Shapiro-Wilk test, $p < .05$). Therefore, Wilcoxon signed-rank tests were used to compare the two conditions.

4.5.2 Results

Figure 10 shows the measures for Study 4. There was no significant difference in recall between the two conditions ($W = 192, Z = -0.56, p = .591, r = 0.07$). The differences in precision and phrase-level accuracy were also not significant (respectively: $W = 184, Z = 0.49508, p = .633, r = 0.06; W = 213, Z = 1.0238, p = .315, r = 0.132$). The average length of the audio clips was 2.1s (SD=0.01) in the *Default* condition

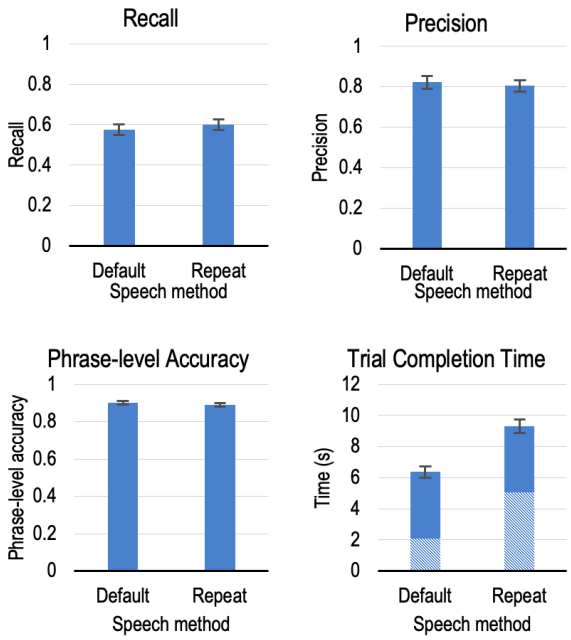

Figure 4.7: Graphs of precision, recall, and phrase-level accuracy in Study 4. The shaded portion in trial completion time indicates the average length of audio clips in that condition. The only time was significantly different between the two conditions. The error bars are standard errors ($N = 30$).

and 5.1s (SD=0.02) in the *Repeat* condition. Due to the longer length of audio, trial completion time in the Repeat condition was also longer than in the Default condition, at 9.3s (SD=0.4) compared to 6.4s (SD=0.4); the difference was significant ($W = 1, Z = -4.76, p < .001, r = 0.61$).

4.5.3 Summary and Discussion

Listening to audio clips twice did not improve the accuracy measures, although it added length to the audio clip. However, this result should be considered to be a preliminary exploration of the repetition approach, with more work needed to evaluate potential interactions with speech speeds, pauses between words, and repetition.

4.6 Discussion

Combined, the four studies show, first, that identifying speech recognition errors through audio-only interaction is hard: participants missed identifying over 50% of errors in Study 1, the majority of which included speech sounds strung across multiple words (*e.g.*, one word recognized as two separate words that sound similar to the original word). Studies 2 to 4 then explored three straightforward speech output manipulations, showing that adding even an imperceptibly brief pause (1ms) between words increases recall and phrase-level accuracy. In terms of speech rate, a high speech rate of 300 WPM reduced the ability to identify errors compared to slower and more subjectively comfortable rates. Finally, repeating the audio output (*i.e.*, playing it twice instead of once) did not impact error identification, at least at a 200 WPM speech rate.

Designing Audio-Only Speech Input. The manipulations we evaluated all add time to the audio output. One design choice would be to add very short inter-word pauses to all dictated text output. However, it may be preferable to provide user control over whether to achieve higher input accuracy at the cost of this extra time. Users could listen to the text output using typical speech settings (*e.g.*, no pause), then if they detect the possibility of an error, they could review the text again in more detail using pauses and slower speech. Depending on the speech recognizer's accuracy, in fact, this could be overall the most efficient interaction style, achieving both high speed and text input accuracy. More work is needed. Users may also have individual preferences regarding the tradeoff between speed and text input accuracy, where some may be more concerned than others about missed errors. The level of concern will also vary based on task context,

similar to how the acceptability of handwriting input recognition errors varies based on context [105]. For example, sending an informal text message to a spouse likely does not require the same level of attention to accuracy as writing an email to one's work supervisor.

Previous work has shown that experience with synthesized speech impacts comprehensibility (*e.g.*, [95]). As such, it will be important to investigate how experience may interact with the effectiveness of pauses, speech rate, and repetition on audio-only error identification. For example, users with visual impairments who are experienced with screen readers, will likely perform differently than the sighted users included in our study. Another factor that could impact the ability of users to identify errors when reviewing audio is the attentional demand in many settings where speech is used, such as mobile interaction [106]. Related, while we did not see an impact of background noise level on error identification rates in Study 1, we hypothesize that our background noise was simply too quiet and that louder noise would cause lower identification rates.

Limitations. Our study has limitations that should be addressed in future work. First, though we purposely chose a single-factor design for Study 3 to bolster statistical power, the follow-up data collection showed that there is an interaction effect between pause length and speech rate, an interaction that should be examined further. Second, we used a transcription task where participants dictated a presented phrase. This approach allows for precise measurement of error identification rates (by comparing errors and participant responses to the presented phrases) but is less realistic than a free-form dictation task would be. Third, we reused the phrase set and errors from Study 1 for all subsequent studies. It will thus be important to generalize the findings to different phrase sets. Fourth,

participants were only provided with feedback on whether they had correctly identified speech recognition errors during the practice trials in Studies 2-4 but not during test trials. It is possible that such feedback, while not representative of real use, would have impacted performance and subjective responses. Finally, while we did not observe any impacts due to the quality of the synthesized speech, future work should examine the potential impacts of different types of speech synthesis engines on error identification.

4.7 Conclusion

We reported on four studies to characterize and address the difficulty of identifying speech recognition errors when using audio-only speech input. Study 1 revealed that by listening to audio clips alone, users could identify less than half of the speech recognition errors. We then addressed the most common type of error that participants had missed in Study 1—errors where multiple words blended together—by inserting pauses between each word and varying speech rate in the audio output. The simple solution of inserting even a 1ms pause between words improved the ability to identify errors, while a fast speech rate made the task more difficult, and repeating the audio output had no effect. These findings have implications for speech-based text input for a variety of non-visual contexts, and an important avenue of future work will be to extend the investigation to accessibility for blind and visually impaired users.

Chapter 5: Comparing Error Identification in ASR Across Blind and Sighted Users

5.1 Motivation and Introduction

The study in Chapter 4 quantified and characterized the challenge of identifying ASR errors. Studying sighted users, it showed that participants missed about 50% of ASR errors when reviewing dictated text with no visual output and standard text-to-speech synthesis (at 200 words per minute). Despite the importance of accurate audio-only speech input for blind users—for example, blind users make use of speech input at higher rates than sighted users [13]–the ability of screen reader users to identify ASR errors has not been evaluated. Individuals with visual impairments who use screen readers are known to comprehend synthesized speech better than sighted people [95]. This leads to the following research questions: *How do blind and sighted individuals' experience with speech input and concerns for ASR errors differ? How well can blind screen reader users identify ASR errors when using speech input?*

In this chapter, we report on an exploratory user study that compares blind and sighted users' experience with speech input and interactions with ASR errors to better understand challenges and strategies in reviewing ASR errors through audio-only. Study

sessions included a semi-structured interview on experiences with speech dictation and synthesized text-to-speech output, followed by a speech dictation task where participants were asked to identify ASR errors in their dictated text. We found that while both user groups confirmed the importance of speech input, blind participants used speech input more frequently than sighted participants (confirming results from [13]). Other differences between the two groups included the most common uses of synthesized speech (reading text on a screen for blind participants vs. conversational interfaces such as Siri for sighted participants) and methods to review the inputted text (visual magnifier[1] or audio for blind participants vs. visual review for sighted participants).

During the initial interview, most participants reported that identifying ASR errors is not challenging, but the performance data in our study suggests otherwise. In the speech dictation task, participants in both groups were only able to identify around 40% of ASR errors in the speech dictation task, and, counter to our hypothesis, there were no significant performance differences between the two user groups. While the challenge of identifying ASR errors through audio-only has been identified for sighted users in a study in Chapter 4, sighted users can choose to review important text visually when needed. That audio-only identification of ASR errors is equally challenging for blind users with substantial synthesized speech experience—and who do not have the option for visual review—emphasizes the importance of developing speech input techniques that more accurately allow blind users to review and edit dictated text.

We show that identifying ASR errors with audio is even more difficult for longer texts, indicating that the length of the message may need to be considered when designing

[1]Our blind participants included two legally blind individuals who used a magnifier to read text.

interfaces for reviewing the dictated text. Based on the analysis of the audio recordings, we found that blind participants dictated their messages slower than sighted participants, perhaps compensating for system limitations, though this difference was not reflected in the corresponding ASR errors. Similarly, we observed that shorter words were used on average for longer messages yet more ASR errors were observed. Most importantly, we identified three distinct strategies that participants used to indicate ASR errors in the played back messages that could lead to novel interactions for reviewing ASR errors: pointing to a specific word(s), indicating the location of the errors in the message, and counting overall errors that they spotted.

5.2 Method

To compare blind and sighted users' experiences with speech input and their ability to identify ASR errors with only audio output, we recruited 24 participants and conducted a two-part study that included a semi-structured interview followed by a speech dictation task.

5.2.1 Participants

We recruited 12 blind participants (6 male, 6 female) who were screen reader users and 12 sighted participants (5 male, 7 female) from campus email lists and local organizations. The sample size was in line with typical sample sizes in this community and designed to balance research goals with practical issues of recruitment and burden on the participant community [103]. Blind participants ranged in age from 23 to 67 ($M =$

ID	Age	Gender	Visual impairment	Age of onset	ID	Age	Gender
B1	33	F	Total blindness	27	S1	26	F
B2	40	M	Light perception	35	S2	22	M
B3	30	F	Legally blind	23	S3	22	M
B4	65	M	Total blindness	Birth	S4	20	M
B5	52	F	Total blindness	15	S5	19	F
B6	59	F	Total blindness	1	S6	27	M
B7	63	M	Light perception	40	S7	19	F
B8	23	F	Total blindness	13	S8	19	F
B9	49	F	Total blindness	34	S9	21	M
B10	54	M	Legally blind	6 months	S10	20	F
B11	67	M	Legally blind	Birth	S11	19	F
B12	64	M	Legally blind	50	S12	31	F

Table 5.1: Participant characteristics, with "B" denoting blind and "S" sighted participants. All but B10 and S12 were native English speakers; B10 and S12 had lived in the US for 30 and 27 years, respectively.

$49.9, SD = 15.1$) and sighted participants were 19 to 31 years old ($M = 22.1, SD = 3.9$). Blind participants reported being totally blind ($N = 6$), having some light perception ($N = 2$), or being legally blind ($N = 4$). All but two participants (one blind and one sighted) were native English speakers[2]. Background information for all participants is shown in Table 5.1; blind participants are denoted "B#" and sighted participants are denoted "S#."

Our blind participants were all familiar with synthesized speech since it serves as speech output for their screen readers; participants used a screen reader several times a day (N=11) or several times a week (only B11). Only one participant across both groups (B12) reported some hearing loss[3].

[2]However, the two non-native English speakers (B10, S12) were not found to be outliers in terms of message length, ASR errors, or missed errors on the speech dictation task, with outliers at 1.5 times the interquartile range [107]. Thus, their data are included in the analysis.

[3]However, we did not find B12 to be an outlier in terms of message length, ASR errors, or missed errors on the speech dictation task, with outliers at 1.5 times the interquartile range. Thus, B12 was also included in the analysis.

5.2.2 Procedure

Study sessions took up to 1.5 hours and were conducted in a quiet room. The whole procedure was video recorded for later analysis of participants' input in the interview and speech dictation task. The session started with a questionnaire to collect demographic information and experience with a screen reader.

Semi-structured Interview. We then conducted a semi-structured interview (30 minutes) on prior experience with synthesized speech, speech input, and ASR errors. For the questions about ASR errors, we defined the speech recognition errors as texts recorded incorrectly by the device because it misunderstands a word or words that the user said. Specifically, participants responded to questions about:

- frequency of use, usefulness, devices, and applications for synthesized speech output
- frequency of speech rate adjustment and reasoning behind these adjustments
- frequency of use, usefulness, devices, and applications for speech input
- maximum length for previously dictated text, and reviewing practices for dictated text
- frequency of encountering and fixing ASR errors
- ASR error importance and how that relates to specific situations
- strategies for identifying and fixing ASR errors

For the two questions regarding the frequencies of using speech input or synthesized speech, frequencies were measured in an *absolute* 7-point scale adopted from Rosen *et*

Figure 5.1: Study setup for the speech dictation task, showing researcher (left) and participant (right) perspectives. The screen was blank across all participants to control for access to visual information.

al. [108] (Never, Once a month, Several times a month, Once a week, Several times a week, Once a day, Several times a day). For example, the absolute scale was used when asking "How often do you use speech input to dictate text?"

Another four questions which were relative to the frequency of using the speech input or synthesized speech employed a *relative* 6-point scale (Never, Very rarely, Rarely, Occasionally, Very frequently, Always) [109]. For example, a question with the relative scale asked, "How often do you encounter speech recognition errors when you dictate text?"

Speech Dictation Task. Participants then completed a speech dictation task using our custom experimental testbed built for the Apple iPhone 8 and using iOS's built-in ASR[4] and synthesized speech[5] features. A female voice with 175 words per minute (WPM) speech rate was used for the synthesized speech. The study setup is shown in Figure 5.1. We employed the free-form text entry task (*i.e.*, composing the text for speech input by a participant) instead of asking participants to read reference text. The free-form

[4] `https://developer.apple.com/documentation/speech`

[5] `https://developer.apple.com/documentation/avfoundation/speech_synthesis`

text entry task is more realistic than reading reference phrases for the speech input because people usually compose a text rather than reading a reference text when they use speech input. Moreover, the free-form text entry task allows us to recruit blind participants from the general population without restrictions on Braille literacy. If the reference phrases had been given to the blind participants in Braille for this task, the participants would have to be Braille readers who are from around 10% of all people with visual impairments [110].

The task consisted of four practice trials followed by 30 test trials. For each trial the participant composing short text or email messages in response to a series of prompted scenarios, then reviewing the recognized text to identify any ASR errors. The overall task description was as follows:

> ”In this task, you will be given a series of situations in which you need to compose a text message or email. For each situation, you will listen to a description with a chime sound at the beginning, then dictate a short text message or email with 1-2 sentences in response.”

The 30 different prompts for the test trials were presented in random order. The test prompts were selected from a list of short scenarios (”situations”) studied by Vertanen and Kristensson [111] for a freeform text composition task, such as: *”Your housemate has been sick for the last week. You are currently shopping downtown. See if he requires anything.”* We asked participants to limit the dictated messages to 1-2 sentences so that they would remember their original input easily when it came time to review for ASR errors. Participants were allowed to make up names for message recipients when desired.

As shown in Figure 5.1, the screen was blank throughout the task so that neither blind nor sighted participants received visual feedback. After completing the 30 trials with short scenarios, the testbed presented three additional trials (prompts for narrative writing from New York Times [112]) with open question prompts that were intended to elicit longer descriptive answers, such as: *"You are filling out an online questionnaire about customer reviews of products. Describe how much you trust online reviews and why."* In these three trials, participants were given no length limit for their dictated messages.

At the start of each trial, the testbed played a chime sound followed by an audio recording of the prompt description. We chose to use pre-recorded audio spoken by a native English speaker for all the prompts to control for any potential effect of synthesized speech for this description on participants' ability to later identify ASR errors through synthesized speech. Participants were allowed to repeat the prompt multiple times to ensure that they understood it and were ready to dictate a response. Participants then double-tapped on the iPhone screen, dictated their message, and double-tapped again to end the dictation. Sound effects played to provide feedback when the system started and stopped recording (the on/off sounds used for Siri on iOS), to help participants speak only while the ASR was activated. Immediately following dictation, the text recognized by the ASR system was played using synthesized speech. After listening once to the synthesized speech output, participants were asked to verbally report any difference(s) between the original speech they had dictated and the text they heard via the synthesized speech output. Participants also reported how certain they were that they had identified all errors in the message by using a 4-point scale (very certain, certain, uncertain, very uncertain). Participants were allowed to redo the dictation for a trial once and only once

if they felt they had made a mistake while speaking (*e.g.*, stumbling over words). Of all participants with 720 trials in total, S5, B1, and B12 opted to re-dictate their input in 1, 1, and 6 trials with short scenarios, respectively. Only one of these instances (a trial of B12) occurred after the synthesized speech output had played. An additional three trials with short scenarios for B5 were redone because the participant's accidental input caused the system to prematurely end the trial. B12 also re-dictated the input in one trial with open questions while speaking in the first attempt[6].

Post-study questions. At the end of the study, we asked questions about the overall experience of reviewing the dictated message during the task. Specifically, participants reported their agreement to the following statements by using a 5-point scale (strongly agree, agree, neither agree nor disagree, disagree, strongly disagree) from Rosen *et al.* [108]:

- "The system correctly recognized almost everything I said."
- "It was difficult to identify errors made by the speech recognition system."

Open-ended questions were used to obtain a rationale for their responses as well as feedback on strategies and challenges in identifying errors.

5.2.3 Measures and Data Analysis

The responses from the participants in the semi-structured interview and speech dictation task were transcribed from the videos of the user study and used to analyze the results. We logged the timing of speech input and the ASR results from the experimental testbed.

[6]The trials where participants re-dictated their input were not found to be outliers, with outliers at 1.5 times the interquartile range. Thus, these trials were included in the statistical analysis.

5.2.3.1 Semi-structured Interview

We qualitatively coded the responses of open questions using a thematic coding method to identify the major themes in the participants' responses [113]. Two researchers collaborated to code the interviews. The first researcher transcribed all of the interview data. The second researcher prepared the initial codebook based on transcription and coded the answers. The first researcher then conducted a peer review of the codebook and of randomly selected transcripts from two blind and two sighted participants. There were 10 disagreements out of 72 coded answers. The two researchers then resolved the disagreements through consensus and updated the codebook with 132 codes for 16 open questions to include two new codes about why participants were using synthesized speech and the method of reviewing text from ASR. Answers for all Likert-scale questions were analyzed with the Mann-Whitney U test, a non-parametric test that allows us to compare ordinal data from the two participant groups.

5.2.3.2 Speech Dictation Task

The speech dictation task used a mixed factorial design with a within-subjects factor of *Prompt* (short scenarios vs. open questions) and a between-subjects factor of *Vision* (blind vs. sighted). To analyze ASR errors from the speech dictation task, we manually transcribed the participants' original speech input and the verbal report of the ASR errors from the video recordings.

The differences between the manually transcribed speech input and the ASR results recorded by the experimental system were considered to be ASR errors. We defined

an *error instance* as an ASR error with a word or a group of consecutive words. Error instances were coded based on their identification by participants as one of three levels of correctness:

- **Identified:** a participant identified mentioned the specific incorrect word(s). For example, if the original input is *"Can I have the vendor's price lists?"*, the ASR result is *"Can I have the vendor's price list?"* (*i.e.*, missing an 's'), and the participant says, *"I said lists instead of list."*, then an identified error instance is "lists." Error instances with multiple consecutive words were considered to be identified if at least *one* of those words was identified exactly, based on the assumption that users would be able to locate that error instance if they wanted to edit it.

- **Noticed:** an error instance was noticed by the participant but was described with some ambiguity. If the participant says, *"I think there was an error in there."* in the above example, "lists" is a noticed error instance.

- **Missed:** a participant did not notice any of the misrecognized words or error instances.

Based on the coded errors, we computed precision (when a participant thinks they identified an error, how often is it actually an error) and recall (the proportion of error instances that participants were able to identify). We measured the WER of the ASR results to see how frequently errors occurred. The length of messages was also measured as the number of characters and the number of words in the original speech input. WER, recall, and precision did not violate the normality assumption (Shapiro Wilk test, $p >$

.05) and were analyzed using Welch's t-tests ($\alpha = 0.05$). Message length violated the normality assumption for sighted participants (Shapiro Wilk test, $p = .023$), so we used a Mann-Whitney U test, a non-parametric alternative to the t-test, for this measure.

We looked into how participants reported the ASR errors during the speech dictation task from the transcribed data. The strategies of reporting errors would be potentially related to how people identify and remember the ASR errors while they are reviewing an ASR result. We found three distinct strategies that participants employed to report the ASR errors on the short scenario trials:

- **Finding a specific word(s):** an error instance was pointed out with the specific incorrect word(s). For example, a participant reported errors by saying "I think there was one error where it missed the word 'the'", "last word it said 'think' instead of 'thinking'."
- **Indicating the location:** an error instance was indicated by its location in the text. For example, a participant said *"I think the last part is messed up [...]"* in this case.
- **Counting:** a participant counted the errors in ASR result (*e.g.*, "I heard two errors.").

The strategies were not used exclusively; participants used one or more than one method in a trial. A total of 274 error instances from 2 blind and 2 sighted participants (randomly selected) were independently coded by two researchers for interrater validation. There was a substantial agreement in the level of correctness (Cohen's kappa[7] =0.75) and almost perfect agreement in the strategy of reporting errors (Cohen's kappa=0.83) [115].

[7]Using cohen.kappa from R package 'psych' [114].

After the validation process, one of the two researchers coded the error instances from all participants.

5.3 Results

5.3.1 Insights from Semi-Structured Interview

The main themes from the interview included experience with synthesized speech and speech input as well as strategies for detecting ASR errors.

5.3.1.1 Experience with Synthesized Speech

While 11 out of 12 blind participants reported using their screen readers several times a day, when asked about the frequency of use for synthesized speech only 9 participants reported several times a day. We suspect that the other 3 participants might have not associated the term ”synthesized speech” with their screen reader voice when answering this question. Still, blind participants reported using synthesized speech more frequently than sighted participants ($U = 17.5, p < .001; r = 0.60$); only two of the 12 sighted participants used synthesized speech on a daily basis (Figure 5.2). Participants in both groups reported using synthesized speech with a range of devices, such as a computer, smartphone, tablet, watch, TV, or smart speaker (*e.g.*, Amazon Echo, Google Home). However, while smartphones were the most popular device for both groups, only one sighted participant used synthesized speech on a computer versus 9 of the 12 blind participants. Blind participants also primarily used synthesized speech when using screen readers ($N = 12$), whereas sighted participants used it mostly with conversational interfaces

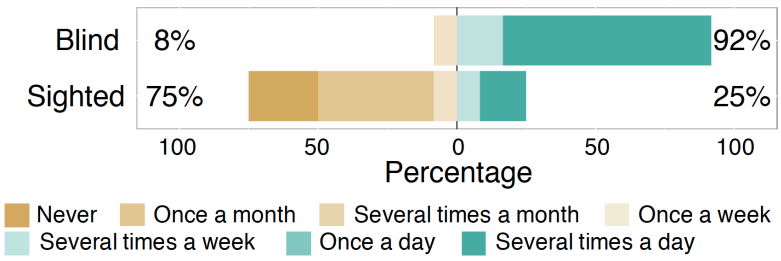

Figure 5.2: Reported frequency of using synthesized speech ($N = 24$).

such as asking Siri a question ($N = 6$) and calling ($N = 3$).

Unsurprisingly, as indicated by prior studies (*e.g.*, [97, 98]), blind participants preferred faster speech rates compared to sighted participants. More than half of the blind participants ($N = 7$) preferred a speech rate setting of 51-100 (around 250-780 WPM [97]) on iOS, which is faster than the default speech rate of 50; the rest preferred the default ($N = 4$) or a slightly slow speech rate ($N = 1$). Blind participants who used faster speech than the default rate were used to listening to fast synthesized speech. Some of the participants mentioned the balance of the comprehensibility and speed. When asked about the speech rate, B3 said *"[...] I think mine is set to something like 57% and basically I can understand everything. If it's faster than that, I may miss some things that it says because it may sound jumbled. If it's slower than that, it may be aggravating [...]"* On the other hand, sighted participants were not concerned by the speech rate, saying they did not have any preferred speech rate ($N = 7$) or that they preferred the default ($N = 5$).

Nine out of 12 blind participants had experience adjusting the speed of synthesized speech, while none of the sighted participants did. Only one of the blind participants, B7, reported doing so frequently, using a fast speech rate for standard listening, but slowing it down for books or articles. Other blind participants adjusted the speech rate occasionally

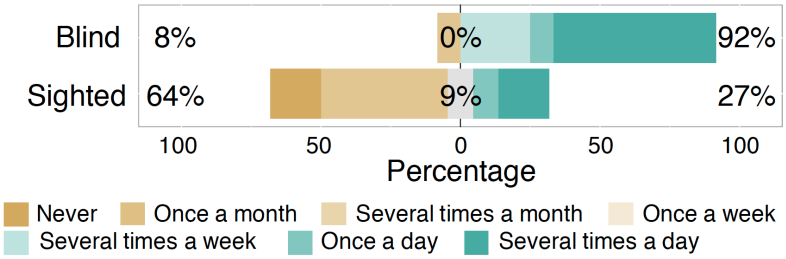

Figure 5.3: Reported frequency of using speech input for dictation and voice commands ($N = 24$).

($N = 4$) and very rarely ($N = 4$) for various reasons: reading certain words or content carefully (*e.g.*, email, books, address), when letting other people use their device to get help or share contents, when getting used to a new device, and just for variety's sake. B2 said, *"[...] If I'm working on someone else's device I would have to adjust their rate to match what my rate is [...] If I'm teaching, I would have to adjust it, so another person could understand because it may be too fast for them [...]"*

5.3.1.2 Experience with Speech Input

Blind participants also used speech input more frequently than sighted participants ($U = 26, p = .006; r = 0.49$), as shown in Figure 5.3 (and confirming [13]). Across both groups, participants most commonly used speech input on a smartphone compared to other devices. In terms of specific tasks, blind participants regularly used speech input for writing a text for various applications ($N = 7$), such as text messages, emails, and filling out online forms while only a few sighted participants used speech input for writing text messages ($N = 4$). It was more comfortable to write texts with speech input than keyboards for blind participants who wrote texts with speech input. B2 said, *"[...] Probably my main reason I mean is really just the convenience of it (speech input) so I*

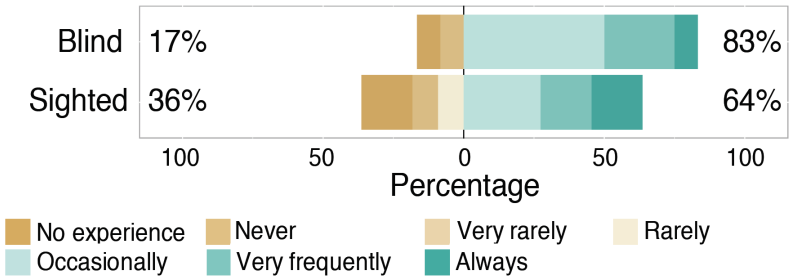

Figure 5.4: Perceived frequency of encountering ASR errors when dictating text ($N = 24$)*

* Participants in "No experience" had not entered text with speech input. Participants in "Never" had entered text with speech input, but never encountered any ASR error.

don't have to really type anything out unless I have to more so the quickness of it." The majority of both blind ($N = 9$) and sighted ($N = 9$) participants used conversational interfaces such as calling, asking Siri questions, opening apps, and setting timers.

Regardless of whether they regularly used speech input for dictating text, to understand differences in how speech input is being used, we asked participants to describe the length of the longest text that they had experience dictating. Of the ten blind participants who had experience dictating text, eight had entered text longer than two sentences and four of those eight had dictated several paragraphs at a time. In contrast, only one sighted participant had dictated an entire paragraph, whereas the remaining eleven reported dictating at most 1-2 sentences.

5.3.1.3 Experience with Detecting ASR Errors

As seen in Figure 5.4, the majority of participants in both groups felt that they encountered ASR errors at least occasionally when dictating text; there was no significant difference between the two groups on this measure ($U = 73, p = .976$). When participants were asked an open-ended question about how concerned they were about ASR errors,

the majority of blind participants expressed deep concerns about ASR errors ($N = 9$) versus only some of the sighted participants ($N = 5$). In particular, B1 said, *"I care about them a lot because I don't want people to think that I'm stupid and I want them to understand what I'm talking about, what I'm trying to say to them."*, highlighting a previously studied misconception on the relation between spelling errors and cognitive abilities such as intelligence and logical ability [95,162]. B10, one of the two blind participants cared moderately, said *"To some extent. I wouldn't say I care extremely or I don't care just as much as I could have it correct." The only blind participant, B3, who care a little said "... [I care] a little because if she can pick up 96% of what I'm saying, I'm happy with that."* No blind participant and four sighted participants reported not being concerned about ASR errors. Those participants did not necessarily feel that ASR was accurate. For example, S12 said, *"I mean I think it's a frustration but it's not a big deal. If it's an informal text it's fine [...] I wouldn't use it [speech input] to write something that's a little more important because it's not as reliable."*

As illustrated by the S12 comment, the importance of ASR errors also varied depending on the situation. To explore such use cases, as a follow-up question we asked participants if there were some situations in which they were more concerned about ASR errors than others. When necessary, we further clarified this question by providing situation themes such as: specific tasks, certain contents, communicating with different people, and being more rushed. Blind participants reported paying more attention when sending a message to someone in a professional relationship such as a work colleague and client ($N = 5$) or in a rushed situation to avoid wasting time in fixing ASR errors ($N = 3$). Blind participants also focused on punctuation marks, certain words that may be likely to be

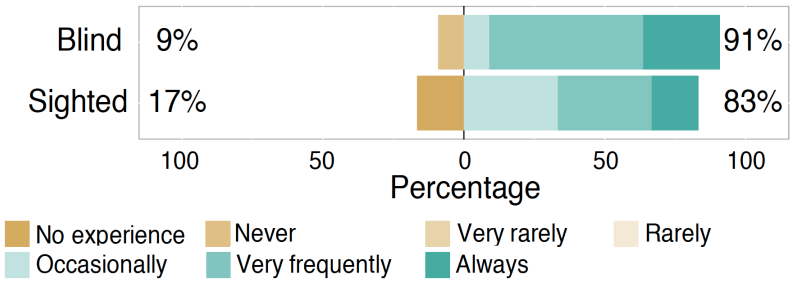

Figure 5.5: Frequency with which participants reported reviewing and editing text after dictation ($N = 24$).

misrecognized by the speech recognizer (*e.g.*, addresses, proper nouns, numbers), and content that may be hard to understand with incorrect speech recognition. B3 said *"[...] if you don't put a period, of course, it's one run-on sentence so again I guess that's user error because if I say period or comma it'll give the space [...]"* B7 said *"[...] I need to speak a person's name, or a location that is something the speech recognition software is very unlikely to recognize and it's essential that the name or location be accurate."* On the other hand, sighted participants said they were most concerned about ASR errors when sending emails to multiple people and when performing a voice search ($N = 6$). Some of the sighted participants ($N = 4$) mentioned rushed situations where they have limited time to review and fix ASR errors. For example, S6, said, *"If I'm more relaxed I don't really care but if I'm rushed and I need to like articulate a text message then I'm going to take the time to actually type it out [...]"* Like blind participants, sighted participants also mentioned concerns about ASR errors when sending a message to a person in a professional relationship compared to family or friends ($N = 6$).

The frequency of reviewing dictated text was not significantly different between blind and sighted participants ($U = 49, p = .168$), as shown in Figure 5.5. Unsurprisingly, blind participants were more likely to review dictated text via audio (synthesized speech

output). Pertaining to the blind participants that reported having reviewed their dictated text ($N = 10$), the majority had used primary audio ($N = 8$) and only (B10, B11) had used audio plus a magnifier. Of the sighted participants, only one (S9) had used audio to review dictated text and did that in conjunction with visual output by listening to the dictated text first, then visually checking if it sounded like there were ASR errors. The remaining sighted participants reported having reviewed dictated text only visually ($N = 9$) or had no experience with speech input for text entry at all ($N = 2$).

Though the study in Chapter 4 showed that the accuracy of identifying ASR errors by audio playback is only around 50%, when asked how difficult it is to identify ASR errors, participants were not aware of such challenge. All participants who had experience with reviewing dictated text thought that identifying ASR errors is not challenging. For example, B2 said *"not challenging at all"*, S2 *"not challenging"*, S9 *"not that hard"*, and B4 *"not really challenging."* Exceptionally, B11 pointed out that some ASR errors are not easy to detect due to the similar sounds with original input: *"[...] you can easily hear an error, but you may not see it, you might not know it's an error. In other words, "to" and it might put two 'o's instead of one or something."* Perhaps, the rest of the participants did not realize the challenge of identifying ASR errors with synthesized speech due to difficulty invalidating what they heard (for the blind participants) or due to limited experience with the audio review (for the sighted participants).

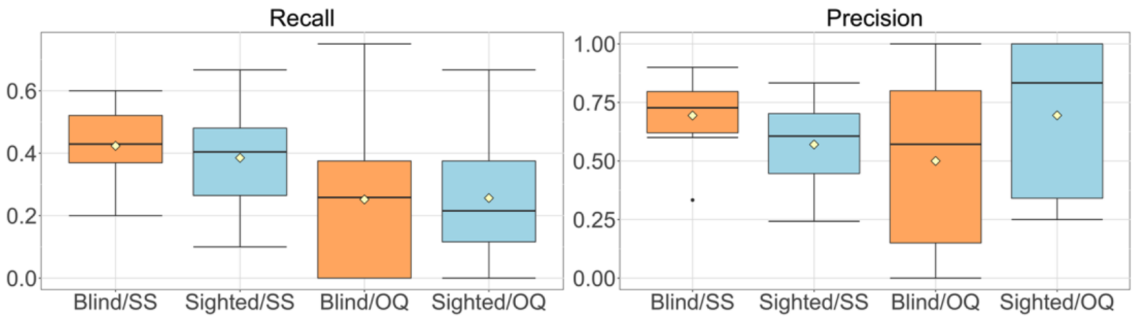

Figure 5.6: Recall and precision for the blind and sighted participants in trials with short scenarios (SS) and open questions (OQ). The trials with open questions had longer messages with higher error rates.

5.3.2 Results from Speech Dictation Task

We report on WER and length of messages, and analyze participants' performance in identifying ASR errors based on precision and recall. Our primary analysis compares blind and sighted participants in the short scenario (SS) trials. As a secondary analysis, we report on the open question (OQ) trials, comparing them to the SS trials. Given that we purposely chose to focus on the SS trials and did not counterbalance the SS and OQ trials, and that there are many more SS than OQ trials, this analysis should be considered exploratory—useful for informing future research directions but not meant to be conclusive. We further analyzed the characteristics of speech input from the participants (*i.e.*, speech rate and length of words) and the error instances (*i.e.*, types of errors and the strategy of reporting errors). The analysis provides the empirical findings of the patterns of entering a text using speech input and identifying errors.

The hypotheses of this task are: (i) blind participants can identify the ASR errors with audio more accurately than sighted participants; (ii) ASR error identification is harder with longer speech input.

5.3.2.1 Differences in Identifying ASR Errors

Figure 5.6 shows the average recall and precision of identifying ASR errors.

Short scenario trials. While prior studies have shown that blind users comprehend synthesized speech better than sighted users [97, 98], this did not translate to a significantly improved ability to identify ASR errors through synthesized speech. Recall, the proportion of error instances correctly identified, was 0.42 ($SD = 0.13$) for blind users and 0.38 ($SD = 0.16$) for sighted users. This difference was not statistically significant ($t_{21} = -0.64, p = .529$). Precision, the proportion of ASR errors identified by the participants that were actually errors (not mistakes on the participant's part), was also not significantly different across the two groups: on average 0.72 ($SD = 0.17$) for blind participants and 0.56 ($SD = 0.20$) for sighted participants ($t_{17} = -1.54, p = .140$).

Open question trials. Compared to the short scenario trials above, identifying ASR errors was more challenging with the three open question trials. The average recall of all 24 participants was 0.25 ($SD = 0.24$), which was significantly lower than SS trials at 0.40 ($SD = 0.15$) ($W = 42, p = .001; r = 0.45$). Specifically, the recall was 0.25 ($SD = 0.24$) and 0.26 ($SD = 0.21$) for blind and sighted participants, respectively in OQ trials. The average precision in open question trials was 0.50 ($SD = 0.40$) for blind participants and 0.69 ($SD = 0.34$) for sighted participants. Average precision of all 24 participants in OQ trials was 0.59 ($SD = 0.37$) in OQ trials and 0.64 ($SD = 0.20$). There was no significant difference in precision between the two types of trials ($W = 91.5, p = .627$).

The most common strategy was finding a specific word(s) which was used by 12 blind participants in 156 trials and 12 sighted participants in 137 trials. Some participants

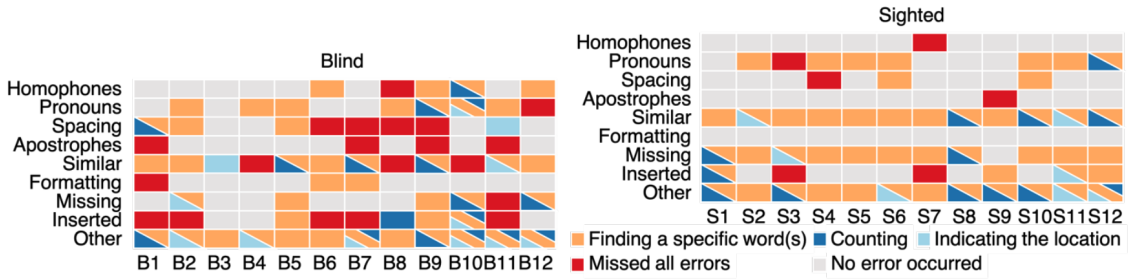

Figure 5.7: The strategy used to report different types of ASR errors by the blind and sighted participants. There is no strategy in a cell if no error occurred or a participant missed all errors.

counted the errors when they identify ASR errors. The eight blind and eight sighted participants had reported the errors by counting the number of errors in 36 and 34 trials, respectively. Nine blind and Eight sighted participants indicated the location of errors in 29 and 18 trials. Figure 5.7 shows that some participants (B6, S4, S5, S7, S8) tend to use the same strategies across different types of errors.

The length of the message and the number of errors in the ASR results may have influenced the strategies. When participants counted the ASR errors or indicated the location of errors, the length of the message was 38.2 ($SD = 24.3$) and 40.6 ($SD = 27.7$) words on average, respectively. The length of the message was only 30.4 ($SD = 19.2$) on average in trials where participants pointed out the specific word to report ASR errors. In trials where participants found a specific word, the ASR results had 1.75 ($SD = 1.5$) error instances on average while there were 2.4 ($SD = 1.9$) and 2.4 ($SD = 2.0$) error instances on average in trials where participants counted or indicated the location of words.

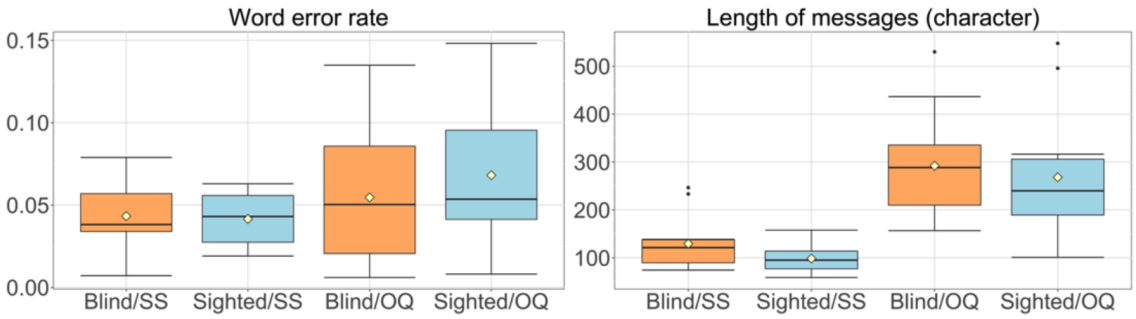

Figure 5.8: WER and length of dictated messages for the blind and sighted participants in trials with short scenarios (SS) and open questions (OQ). Participants dictated longer messages in trials with OQ than SS. There was no significant difference in WER between sighted and blind participants.

5.3.3 Characteristics of Dictated Messages

There are many characteristics of the dictated messages that could relate to the number of ASR errors that participants were able to identify. To better contextualize our findings we report differences in word error rate, message length, speech rate, and word length across the recordings of blind and sighted participants as well as across short scenario and open question trials.

Word Error Rate and Length of Messages. Overall, no significant differences were found in WER or message length for the two user groups. Figure 5.8 shows the average WER and length of messages.

Short scenario trials (SS). In the 30 SS trials, the average WER of blind and sighted participants' speech input was 0.04 ($SD = 0.02$) and 0.04 ($SD = 0.02$), respectively, which is similar to the WER of state-of-the-art ASR engines [7]. We asked participants to keep their dictated messages to 1-2 sentences in length. Blind and sighted participants' dictated messages were 129.3 ($SD = 56.8$) and 98.5 ($SD = 29.5$) characters, which were 25.9 ($SD = 11.4$) and 19.9 ($SD = 6.1$) words, respectively; the medians were 121.3

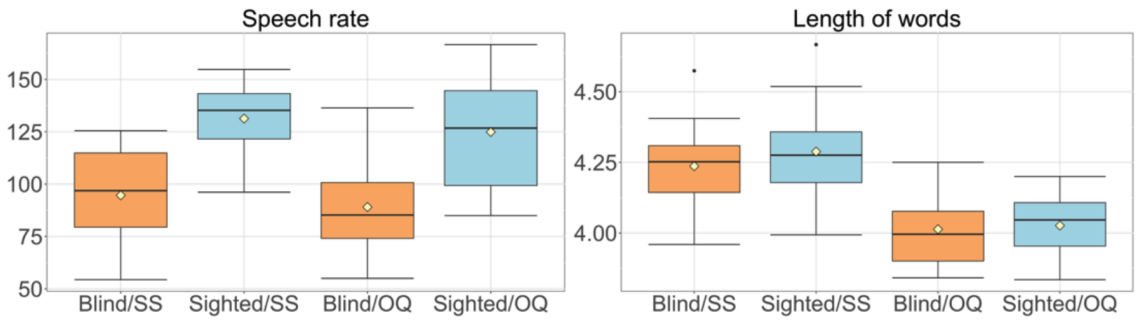

Figure 5.9: Speech rate and length of words for the blind and sighted participants in trials with short scenarios (SS) and open questions (OQ). Blind participants spoke slower than sighted participants. The average length of words was shorter in OQ trials than SS trials.

characters (24.7 words) for blind participants and 95.1 characters (19.1 words) for sighted participants; this difference was not statistically significant (calculated in character; the mean ranks of blind and sighted participants were 14.4 and 10.6, respectively; $U = 49, Z = 1.33, p = .198$).

To examine if the characteristics of the ASR results impacted the accuracy of identifying errors, we compared the trials with and without errors in terms of the message length and the number of errors in a trial. The average number of words was 32.7 ($SD = 20.4$) in trials with missed errors and 24.8 ($SD = 15.0$) in trials without missed errors. The average number of errors was 3.6 ($SD = 2.2$) in trials with missed errors and 2.2 ($SD = 0.8$) in trials without missed errors. The result shows that the length of the message and the number of errors in the ASR result are potential factors that would affect the accuracy. That is, users would be able to identify ASR errors better in a shorter message and when there are fewer ASR errors.

Comparing SS and OQ trials. As expected based on the task instructions, the dictated messages were longer for the OQ trials than the SS trials, at on average 292.1 ($SD = 108.1$) 268.1 ($SD = 134.4$) characters, which were 56.2 ($SD = 21.6$) 51.3

($SD = 25.5$) words, for blind and sighted participants, respectively. The average length of messages of all 24 participants in OQ trials was 280.1 ($SD = 120.0$) characters (53 words, $SD = 23.5$) which were longer than SS trials at 113.9 ($SD = 46.9$) characters (22.9 words, SD=9.4); this difference between SS and OQ trials was significant (calculated in character; $W = 0, Z = -4.29, p < .001; r = 0.62$). Average WER of all 24 participants was also significantly higher in the OQ trials at 0.06 ($SD = 0.04$) than the SS trials at 0.04 ($SD = 0.02$) ($t_{23} = -2.63, p = .015; d = 0.60$).

Speech Rate and Length of Words. We analyzed the original speech input of the short scenario trials from the blind and sighted participants in terms of the speech rate and the length of words to examine if the experience with speech input influences speech rate or complexity of words in the speech input.

Figure 5.9 shows the speech rate and length of words. In the SS trials, blind participants spoke slower than sighted participants with 94.6 ($SD = 22.9$) and 131.3 ($SD = 17.2$) WPM speech rates, respectively ($t_{20} = 4.42, p < .001, d = 1.81$). However, we did not observe this to be reflected on the WER of ASR. As shown in the previous section, there was no significant difference between blind and sighted participants. In the comparison between SS and OQ trials, there was no significant difference in the speech rate ($t_{45} = 0.72, p = .476$). The blind and sighted participants spoke at 113.0 ($SD = 27.3$) WPM speech rate in SS trials and 107.0 ($SD = 30.5$) WPM in OQ trials on average.

The average length of words in the speech input was 4.2 characters ($SD = 0.2$) for blind participants and 4.3 characters ($SD = 0.2$) for sighted participants in SS trials with no significant difference ($t_{21} = 0.22, p = .832$). On the other hand, the average length

of words in SS trials from all participants at 4.3 ($SD = 0.2$) characters were longer than the length of words in OQ trials at 4.0 ($SD = 0.1$) characters ($t_{40} = 5.06, p < .001, d = 1.46$). Considering that the OQ trials had higher WER than the SS trials, speaking shorter words would not have a positive impact on reducing ASR errors.

5.3.3.1 Error Analysis

In the SS trials, there were 340 error instances for blind participants and 236 ASR error instances in total for sighted participants. Participants in both groups missed more than *half* of ASR error instances: missed error instances represented 52.1% ($SD = 11.1$) and 52.1% ($SD = 12.2$) of all error instances on average for the blind and sighted participant groups, respectively. A further 42.3% ($SD = 12.7$) for blind participants and 38.5% ($SD = 16.5$) of ASR error instances for sighted participants were *exactly* identified. Finally, only a small portion of errors, 5.6% ($SD = 4.7$) for blind participants and 9.4% ($SD = 12.5$) for sighted, were *noticed*.

To further understand error identification challenges, we assessed what types of ASR errors were missed in SS and OQ trials. As expected, based on past work [13], participants missed ASR errors when the errors sounded like the original words as shown in Table 5.2. However, not all missed errors related to similar-sounding words as shown in Table 5.3. In general, the accuracy of identifying the error instances with similar sounds with original words was lower at 36.8% than the accuracy of identifying error instances that did not sound like the original words at 44.8%.

Though the sound of error instances in 'spacing', 'homophones', and 'apostrophes'

Type	Description and example	Occured	Identified
Pronouns	Pronouns were recognized as other words with similar sound (*e.g.*, 'Carol' and 'Cara').	54	64.8% (35/54)
Spacing	The recognized words were incorrect because of the spacing (*e.g.*, 'prototype' and 'proto type').	15	33.3% (5/15)
Homophones	The recognized words were homophones of the original words (*e.g.*, 'owe you' and 'OU').	11	36.3% (4/11)
Apostrophes	The recognized words were incorrect because of an apostrophe mark (*e.g.*, 'doctor's' and 'doctors').	5	0.0% (0/5)
Similar	Recognized words have similar sounds with the original words (*e.g.*, 'I'll' and 'I will').	135	27.4% (37/135)
	Total	220	36.8% (81/220)

Table 5.2: Definition and the number of error instances for the types where error instances sounded like the original words. The identified column includes the proportion of exactly identified errors (the number of exactly identified error instances divided by the number of all error instances)

Type	Description and example	Occured	Identified
Missing	The original word(s) was missed in the recognized text.	46	58.7% (27/46)
Inserted	The error word(s) was inserted in the recognized text though they were not spoken by the participants (*e.g.*, recognized text of filler words).	51	17.6% (9/51)
Formatting	The recognized text has a different format (*e.g.*, '2 o'clock PM' and '2:00 PM').	4	50.0% (2/4)
Others	The recognized words sound differently from the original words. We did not observe a common pattern among these errors (*e.g.*, 'we are' and 'of years', 'I think' and 'Of think').	5	47.7% (123/258)
	Total	359	44.8% (161/359)

Table 5.3: Definition and the number of error instances for the types where error instances did not sound like the original words. The identified column includes the proportion of exactly identified errors (number of exactly identified error instances / number of all error instances)

is almost the same as the original words, participants could identify a few of them. For example, B6 guessed the misrecognition of 'too' as 'to' in a trial, saying *"[...] I think it might've said the wrong version of too."* Some participants picked up the small difference in synthesized speech caused by a space between words. For example, S10 distinguished 'prototype' and 'proto type', saying *"It just said 'prototype' like 'prawto type' do you guys care about how it says words? That's the difference."* Participants identified the errors better when some words are missing in the recognized text than when additional words are inserted into the recognized text. The proportion of identified error instances was higher than 50% for the 'missing' type while it was only 17.6% for the 'inserted' type.

It is hard to identify the error instances of some types that have almost the same sound like the original words (*i.e.*, 'spacing', 'homophones', 'apostrophes') with audio-only. To assess the ability to identify errors that can be distinguished with audio, we measured the precision and recall of SS trials after excluding the error instances of 'spacing', 'homophones', 'apostrophes' types. Still, there was no significant difference in precision and recall between blind and sighted participants ($t_{21} = -2.01, p = .056; t_{21} = -0.42, p = .677$).

5.3.3.2 Subjective Certainty

For each trial, participants were asked how certain they were that they had identified all ASR errors in their dictated text using a 4-point scale (very certain, certain, uncertain, very uncertain). In the short scenario trials, blind participants were confident, being

"very certain" in 247 (68.6%) trials, certain in 104 (28.9%), and uncertain in only nine (2.5%). There was no trial where blind participants were very uncertain. Similarly, sighted participants were very certain in 237 (65.8%) trials, certain in 110 (30.6%), uncertain in 12 (3.3%), and very uncertain in 1 (0.3%). These numbers might not be surprising given that all but one participant, who had experience with reviewing dictated text, reported that they did not think that identifying ASR errors is challenging (Section 4.2).

While participants were confident in more than 96% of trials, the accuracy of identifying ASR errors in those trials was still low in terms of recall (very certain: 0.37; certain: 0.46) and precision (very certain: 0.67; certain: 0.61). Perhaps, this could be explained by the fact that some ASR errors were difficult to detect because the errors sounded like the participants' intended words, as in [13]. Another plausible explanation could be that when interacting with a reliable ASR (with WER around 4% in our study), participants may have been less vigilant and less able at detecting ASR errors when they occur. Prior work, surveyed in [116], indicates that complacency could explain why more reliable automation hurts the identification of system errors.

5.3.3.3 Qualitative Feedback

After completing the ASR dictation task, participants were still positive about the performance of ASR and their ability to identify ASR errors. When participants were asked if they agree that the system correctly recognized their input (5-point scale), 9 blind and 10 sighted participants agreed or strongly agreed; there was no significant difference between the two participant groups ($U = 76, p = .914$). Participants also disagreed

when they were asked if it was difficult to identify ASR errors: eight blind and eight sighted participants disagreed or strongly disagreed in this question. Again, there was no significant difference between the two groups ($U = 90, p = .375$).

When asked about any other difficulties they had during the task, seven blind participants reported no difficulty at all, while the remaining five blind participants mentioned challenges in remembering ASR errors in a long text, checking punctuation marks, and distinguishing words with similar sounds. For example, B3 said: *"I knew there was a mistake in the beginning and the end but anything in the middle was fuzzy because these were like I said random tasks."* Contrastingly, 11 of the 12 sighted participants said they had difficulties, including remembering ASR errors in a long text, imperfect pronunciation of synthesized speech, and the fast rate of synthesized speech. S12 said: *"If there are a couple of little errors in the larger text then you kind of lose track of them. [...] another is, is it me who's like am I creating and I saying it incorrectly or is the system picking it up incorrectly?"*

5.4 Discussion

The semi-structured interview showed differences between blind and sighted participants with respect to their experience with speech input and error identification. In the speech dictation task, blind participants spoke slower than sighted participants when they use speech input. We also found that the length of the speech input impacts the accuracy of error identification. Further analysis of the errors characterized the patterns of identifying ASR errors. The empirical findings from the user study provide some insights for future research.

5.4.1 Implications

Need for accessible ASR error reviewing through audio-only interactions. Our findings reinforce the importance of improving text-entry through audio-only for blind users, confirming past studies that show that blind users are more likely to use speech dictation than sighted users [13, 60]. However, when it comes to reviewing their dictated text, our interview findings show sighted participants used the visual output, which is only available to blind users through the text-to-speech audio. Perhaps this explains why blind participants were more concerned about ASR errors than sighted participants, given the difficulty of reviewing the ASR results through audio. For both groups, context relates to their concerns about ASR errors (*i.e.*, kinds of tasks, content, the recipient of the dictated message, rushed or relaxed situations), suggesting that in some cases, users may be willing to use a more time-consuming but accurate reviewing process than simply hearing back their dictated message.

Mismatch between ability and perception of challenges in finding ASR errors. Neither participant group felt it was challenging to identify ASR errors by just listening to the dictated message. However, when asked to perform this task, they missed more than half of the ASR errors. This contradiction suggests that users may be making more errors than they are aware of in their dictated text motivating future work in assessing real-world error rates in dictated and reviewed messages. Therefore, future work is needed to develop an interface for enabling blind users to check the final text after revision in an efficient way rather than going through the text letter by letter.

Higher chance of missing errors with longer text. Comparing the results from the

trials with short scenarios to the trials with open questions also showed that identifying ASR errors is more difficult with longer input. With longer input and higher WER in the trials with open questions, participants had to identify more errors with the long text than the short text. This would have increased the mental load of the task, requiring participants to remember more ASR errors. Since blind participants were more likely to have experience dictating longer passages of text than sighted participants, this challenge may unduly affect blind users. It will be important to consider whether mechanisms to support users in reviewing and editing speech dictation via synthesized speech output will need to differ for shorter versus longer passages of text, such as supporting users in reviewing only one sentence at a time.

Little impact of experience with a screen reader on the ability to find ASR errors. Contrary to our hypothesis, no significant differences were found between blind and sighted participants' ability to identify ASR errors through a synthesized speech on our speech dictation task. Though our interview showed that the blind participants had more experience than sighted participants in reviewing dictated text via audio, only two blind participants who also used magnifiers had the opportunity to confirm what they heard through a synthesized speech by checking visually. This lack of visual confirmation may have led them to overly trust the ASR results as compared to sighted users who had on average substantial exposure to visual feedback of ASR results. The relatively low WERs seen in the task, though reflective of state-of-the-art automatic speech recognizers [7], may have also made it more difficult to detect a statistically significant difference between the two user groups.

Distinct strategies for reporting ASR errors can lead to novel interactions. We

found three distinct strategies of identifying ASR errors by analyzing how participants reported the ASR errors during the speech dictation task. The most common strategy was finding a specific word(s) of the ASR errors. The other two strategies are indicating the location of errors and counting the errors. We found that the average length of messages was shorter and the number of errors was fewer in trials where participants found a specific word(s) than the trials where participants counted or indicated the location of errors.

The selection of strategy would be potentially related to the length of the message and the number of errors in the ASR result. When the text is long or the ASR result includes many ASR errors, participants would have counted or remember the location of errors rather than memorizing words to reduce the mental load. The future study on designing the accessible interface of reviewing ASR results needs to consider that the strategy of identifying ASR errors can be influenced by the length of the message and the number of ASR errors.

Variation of speech input in different contexts. The analysis of speech rate provides empirical evidence that blind users speak slower than sighted users when they enter a text with the speech input. The difference in speech rate would have been caused by blind participants' caution to avoid the ASR errors. A prior study showed that a user articulates the speech when they want to enter a text with speech input without errors [55]. In this case, blind participants would compensate for potential limitations of the ASR system by speaking slowly.

5.4.2 Limitations

The speech dictation task in the user study was designed to make it realistic to the participants by employing the free-form text entry task. Though we were able to measure the ability to identify ASR errors and characterize the use of speech input, the design also has some limitations.

Limitation of the free-form text entry. For our speech dictation task, we employed a free-form text entry task that allows participants to compose texts for themselves. Though the study in Chapter 4 evaluated the ability to identify ASR errors using reference phrases, the free-form text entry was adopted in this article because of the advantages mentioned in Section 5.2.2. However, the user study design with the free-form text entry has a few drawbacks compared to using reference texts. The free-form text entry task can result in ambiguity during error coding by the research team given that the team has only access to the spoken messages by the participant and not the ground truth text phrases. For example, some proper nouns were accurately recognized by the ASR engine (*e.g.*, city and product names) but others were ambiguous (*e.g.*, did the user intend to spell the name Steven or Stephen). In these cases, if the proper noun in the synthesized speech has correct spelling, then we marked it as correct. However, the proper noun (*e.g.*, 'Barbara') that was recognized as a common noun(s) (*e.g.*, 'barber') has been considered as an ASR error. The participants dictated 13.6 proper nouns on average throughout the task.

Missing the semantic change in metrics of performance. In this work, we analyzed the performance of an ASR system for text entry through speech only both in terms of WER but also in terms of participants' ability in identifying these errors using metrics

such as recall and precision. A limitation of these metrics is that they focus on the number of error instances instead of the degree of change in the meaning of the text. For example, if ”want” and ”can” are recognized as ”wanted” and ”can't,” the latter usually changes the meaning of the text more significantly than the former. However, WER, recall, and precision cannot reflect such differences in error analysis [117]. Metrics reflecting the semantic change of the original text due to the semantic differences between ASR errors (*e.g.*, ACE metric by Kafle *et al.* [118]) would also be useful to examine.

Small sample size. The small number of participants in this study limits the statistical power to detect the significant difference with a small effect size, though 24 participants is a common sample size in the CHI and ASSETS community. Therefore, in the analysis of precision and recall, this limitation may have resulted in no statistically significant difference in the comparison of blind and sighted participants. The small number of participants also makes the statistical analysis subject to change by potential outliers. Considering this limitation, we conducted another statistical analysis of the data from the speech dictation task where any outliers were excluded. Specifically, there were four outliers (S3, S11, B10, B11) in terms of dictated message length and one outlier (S11) in terms of precision. Removing these outliers did not change any of our results.

5.5 Conclusion

We explored the experience of speech input, synthesized speech, and ASR error identification through a semi-structured interview and evaluated the ability to identify ASR errors through a task of entering and reviewing text using speech-only. From the

semi-structured interview, we found that sighted and blind participants' experiences differ in many aspects such as tasks, devices, and frequency of using speech input as well as employed methods for reviewing the dictated text. Though most participants reported that identifying ASR errors is not a challenging task, participants in both groups identified only around 40% of the ASR errors. This indicates that identifying ASR errors is challenging even for blind users who may have more experience with speech input and synthesized speech compared to sighted participants. We also characterized how participants identified ASR errors through the analysis of the speech input, the ASR errors, and strategies for pointing to ASR errors in the speech dictation task. These findings enable us to better understand and quantify the challenges in identifying ASR errors for both sighted and blind users. More so, they reveal the need for further research on improving user interaction for speech-only text input that relies on inherently error-prone systems such as ASR.

Epilogue to Part I

In Part I of this thesis, the challenge of identifying ASR errors was characterized through crowdsourcing and controlled lab studies with blind and sighted participants. The studies investigated the impact of manipulating the synthesized speech (*i.e.*, inserting pauses between words and repeating the synthesized speech), types of missed ASR errors, the accuracy of identifying ASR errors, and strategies of identifying ASR errors. The controlled lab study also revealed the experience of identifying ASR errors in terms of how much they cared about the errors, different attitudes of reviewing the recognized text in various situations, etc.

The study with crowdsourcing explored the challenge of identifying ASR errors to examine quantitatively whether identifying ASR errors with synthesized speech only is challenging. The user studies were conducted through crowdsourcing where participants were asked to identify ASR errors after listening to the synthesized speech of a phrase. Participants were able to identify only around 50% of errors, showing that identifying ASR errors through audio is challenging. Inserting pauses between words and slow speech rate improve the accuracy. On the other hand, repeating the audio of the synthesized speech does not help users identifying ASR errors accurately.

Next, we conducted a controlled lab study to compare the experience with synthesized

speech and the accuracy of identifying ASR errors between blind and sighted participants. The hypothesis of the study was that blind users care more about the ASR errors and have better accuracy of error identification due to the more frequent experience with the speech input and synthesized speech than sighted users. The results showed that both blind and sighted participants identified only around 40% of the ASR errors though they thought identifying ASR errors is not a challenging task. On the other hand, blind participants cared more about the ASR errors. The analysis of the speech dictation task characterized the strategies of identifying errors when participants review the ASR results.

Part I of this thesis reported on the completed work and answered the following research questions:

- **RQ1**: How frequently are ASR errors missed? (The user studies in Chapter 4 and 5measured the blind and sighted participants' accuracy of identifying ASR errors through synthetic speech. They showed that both blind and sighted participants missed around 50% of the errors.)

- **RQ2**: Do different synthetic speech manipulations affect the user's accuracy of identifying ASR errors? (We found from the results of the user studies in Chapter 4 that synthetic speech with slower speech rate and pauses between words can increase the accuracy of error identification with audio.)

- **RQ3**: For what tasks do blind and sighted users use ASR? (The study in Chapter 5 showed that blind people use ASR mainly for entering texts while sighted people use it for conversational interface or voice commands.)

- **RQ4**: How different are the experiences with speech dictation and listening between

blind and sighted users? (Both blind and sighted participants thought identifying ASR errors with synthetic speech is not challenging but blind participants cared more about the errors as shown in Chapter 5.)

- **RQ5**: Is the accuracy of identifying ASR errors different between blind and sighted users? (The results of the speech dictation task in Chapter 5 showed that both blind and sighted participants missed around 50% of errors with no statistically significant difference.)

- **RQ6**: What are the blind and sighted users' strategies of pointing to ASR errors? (The user study in Chapter 5 found that they spot a specific word, indicate locations where errors occurred, or count the number of errors.)

Part II: Interacting with Error-Prone Image Recognition

Prologue to Part II

In Part II, this work explores the challenge of identifying errors in camera-based assistive apps with blind users and reducing errors in TOR through iterations with blind and sighted users. Identifying and validating the predictions from camera-based assistive apps would be even hard for blind people because they depend on the visual characteristics of the inputs. While prior studies have shown that blind users actively use their cameras in mobile devices for fun, preserving memories, using social media, and using assistive apps, many of the studies focused on the challenges in blind photography and developing a user interface for blind photography. In this work, we characterize the challenges of validating the predictions from the camera-based apps and identifying the errors. Furthermore, we investigate the usability issues of TOR with blind and sighted users, considering another perspective in using an image recognition system, personalizing it with a teachable interface.

While training and validating a machine learning model (*i.e.*, machine teaching) has been mostly conducted by experts in machine learning, recent intelligent systems enable end-users to conduct the machine teaching task to personalize the system for their idiosyncratic environments and inputs. (*e.g.*, teachable object recognizer [20], personal sound detector [119]). Though building a teachable interface where end-users personalize an intelligent system through machine teaching is technically possible with few-shot

learning or meta-learning approaches, end-users with little knowledge in machine learning would have difficulties in training and validating a machine learning model with their own data samples. In this research, we explore the difficulties from two perspectives: (i) understanding non-experts' patterns and misconceptions in machine teaching (Chapter 8) (ii) designing a mobile TOR app to enable blind users to review their teaching strategies to reduce errors. (Chapter 9)

Part II consists of three studies that investigate blind users' experience with camera-based assistive apps, explore sighted non-experts' challenges in machine teaching, and developing a mobile TOR app for blind users. The first study consists of a semi-structured interview and an error identification task to explore the users' challenges in identifying pre-trained image recognition errors (Chapter 7). In the second study, we recruited 100 sighted participants who are non-experts in machine learning through crowdsourcing. The participants were asked to train and validate TOR through the web (Chapter 8). The photos and feedback from the participants revealed patterns in non-experts' strategies to conduct machine teaching (*i.e.*, how they train an object recognition model, test it, and change their training strategy after observing errors). In the third study with blind participants, we evaluate a TOR app designed based on the findings in the first study and a prior study on the feasibility of using TOR for blind people [20]. The study includes tasks of training, testing the app, and managing the information of objects in the users' datasets (Chapter 9).

Part II of this thesis aims to resolve the following research questions:

- **RQ7**: For what tasks and objects do blind users take photos? (We will examine

RQ7 in Chapter 7.)

- **RQ8**: How did blind users identify the image recognition errors? (We will examine RQ8 in Chapter 7.)
- **RQ9**: What are the blind users' accuracy of identifying the object recognition errors? (We will examine RQ9 in Chapter 7.)
- **RQ10**: What are their strategies for identifying the errors? (We will examine RQ10 in Chapter 7.)
- **RQ11**: What are non-experts' teaching and debugging strategies for a teachable object recognizer? (We will examine RQ11 in Chapter 8.)
- **RQ12**: Do teaching strategies evolve through iteration? (We will examine RQ12 in Chapter 8.)
- **RQ13**: How could descriptors be useful for avoiding errors due to their training examples? (We will examine RQ13 in Chapter 9.)
- **RQ14**: What are blind users' teaching and debugging patterns? (We will examine RQ14 in Chapter 9.)

Chapter 6: Background

This chapter discusses prior work on object recognition focusing on error identification and assistive technologies for blind people. As the work in Chapter 8 and 9 are kinds of machine teaching studies, a research field that investigates people who teach a machine, we present background studies on machine teaching.

6.1 Image Recognition and Error Identification

Object detection and classification have been actively studied for decades as they are fundamental and challenging problems in computer vision. Specifically, object detection aims to provide the location and size of the object instances in an image (*e.g.*, bounding boxes) [120]. The goal of object classification is figuring out whether objects in a set of classes exist in an image or not [121]. Object detection and classification are employed in a variety of applications including blind navigation systems (*e.g.*, [122, 123]), object recognizer for blind people (*e.g.*, [124, 125]), and image captioning (*e.g.*, [126]). In this thesis, we employ a general term, image recognition, that embraces both object detection and classification tasks [127, 128] to indicate general applications related to object detection and classification. Recently, the emerging image recognition systems achieved dramatic improvements with deep learning techniques [129]. However, they are

still error-prone due to the high variations of objects, a huge number of object categories, and limited computing power in mobile/wearable devices [121].

Errors in an image recognition system affect blind users' experience with it significantly as most blind users depend on the output from the system due to the difficulty in verifying the output [130]. For example, a blind user who uses social media rely on the automatically generated captions to understand photos without sighed help and may be confused by the errors in the captions [33]. Therefore, researchers in human-computer interaction and accessibility have emphasized the importance of providing the confidence of the outputs and degree of reliability of AI-infused systems to users [10, 130]. While the impact of object recognition errors on user experiences has not been explored thoroughly, prior studies presented some cases where such errors caused problems. A study on blind users' experience in using social media with images revealed that they overtrust the automatically generated captions even when incorrect captions make little sense [33]. While some errors in blind navigation systems are acceptable when blind users are familiar with surrounding environments, the errors are not acceptable when people around the user react with misguided responses [49]. When image recognition is used to help blind users control household objects (*e.g.*, turning on/off a stove, finding an outlet), errors would cause safety threats. Therefore, such tools are required to have robust safety mechanisms, or users are recommended to use other types of assistive tools such as voice command to control household objects rather than using computer vision-based tools [131]. Given the significance of error handling in using object recognition systems for blind users, this work explores and characterize their challenges in identifying and recovering from object recognition errors.

6.2 Image Recognition for Accessibility

Object recognition has been actively used to develop assistive tools for people with disabilities. For example, body movement recognition has been used to create assistive tools for people with motor impairments such as rehabilitation systems with automatic body movement guidance [132], automatic symptom diagnosis of motor impairments through motion analysis [133], and gesture recognizer as an input method for controlling robots [134] and computers [135]. Assistive tools for people with cognitive impairments also employ computer vision and object recognition. The body motion analysis with images or videos is used to detection of autism-related behavior automatically [136, 137]. Moreover, object recognition and computer vision have been used in assistive technologies for other types of disabilities such as hearing impairments(*e.g.*, [138, 139]) and visual impairments (*e.g.*, [20, 140, 141, 142]). In this thesis, we focus on blind users' experience with camera-based assistive tools.

As object recognition can enable blind people to have a better sense of the visual world, many products that enable people to read texts, distinguish colors, and recognize objects with computer vision are already on the market [124, 125, 143, 144, 145]. Due to the significant impact of the errors in these tools, prior studies have presented guidelines for AI-infused systems with recommendations to enable users to recover from errors [10, 50, 51]. However, they targeted general AI-infused systems including applications for sighted people without an in-depth analysis of blind users' interactions with such systems. A unique aspect of this thesis is that it covers blind users' interactions with both *pre-trained* camera-based assistive tools and *teachable* object recognizers (TOR). Since TOR

was shown to be useful for blind people [24], researchers have been actively developing user interfaces that help blind people to take photos to train an object recognition model [141, 146]. However, as TOR is a kind of emerging technology, it has many issues to resolve such as developing user interfaces for blind users to identify, understand, and recover from errors effectively through machine teaching.

6.3 Machine Teaching and Teachable Interfaces

Machine teaching involves a teacher who knows the decision boundaries and designs an optimal training set for one or more students [4]. In this paper, the teacher is a human and the student is a classification model who is being trained to classify images of objects, as shown in Figure 6.1, though the inverse – machines teaching humans to classify images – is also an active area of research [147]. There is rich literature on sequential machine teaching with humans as the teacher, *e.g.*programming by demonstration for teaching robots to manipulate objects [148, 149]. However, in this review, we focus on prior work that utilizes batch teaching, where examples are given as a set and their order does not matter.

Batch teaching is a very common paradigm for many real-world AI-infused systems, *e.g.*using face recognition, fraud detection, and speech recognition. This is typically done by experts in the field and end-users are hardly exposed to the underlying mechanisms that could help explain their limitations. Teachable interfaces[1] that fall under this machine teaching paradigm, have the potential to help in this direction as they can enable non-

[1] A term coined by Patel and Roy (1998) [150], where "the user is a willing participant in the adaptation process and actively provides feedback to the machine to guide its learning."

Human vs. machine	T=machine, S=machine	T=machine, S=human	**T=human, S=machine**	T=human, S=human
One vs. many	**One student**		Many student	
Batch vs. sequential	**Batch learning**		Sequential learning	
Teaching signal	Synthetic / constructive teaching	Hybrid teaching	**Pool-based teaching**	
Model-based vs. model-free	Model-based teaching	Graybox teaching	**Model-free teaching**	
Student awareness	The student anticipates teaching		**The student does not anticipate teaching**	
Angelic vs. adversarial	**Angelic teaching**		Adversarial teaching	
Theoretical vs. empirical	Theoretical teaching	Hybrid	**Empirical teaching**	

Figure 6.1: Characterization of our testbed in the machine teaching problem space [4], where T stands for teacher and S for student. A human T employs a pool-based, model-free, angelic, empirical teaching. The testbed has a single recognition model S learning in batch mode, unaware that is being taught, while considering T as a friend (no adversarial examples).

experts to uncover basic machine learning concepts (*e.g.* [151]). Moreover, with advances in transfer learning [152, 153], they can spur innovation as end-users can re-purpose models trained on vast amounts of data for new but related tasks, *e.g.*personalize assistive technologies [154].

We look into prior work employing teachable interfaces, a term perhaps not originally used by the authors. Here, we focus on a subset of interactive machine learning literature, where users are called to generate all the training and testing examples for a personalized model. Table 6.1 presents representative examples of prior studies from 2011-2019 on gesture recognition for musicians [155], sign language [156], educational applications [151], personalized sound detectors for people who are deaf/Deaf or hard-of-hearing [119], personal object recognizers for blind people [20], and physical activity classifiers for young athletes [157]. In contrast to this work, prior studies tend to have smaller participant

Table 6.1: Related studies' characteristics juxtaposed with ours.

		[155]	[156]	[119]	[20]	[151]	[157]	This study
Setting	People	1,7,21	10	12	8	30	5	100
	Controlled	•	•	•	•	•	•	
	Real-world	•						•
People	Crowd							•
	Children					•	•	
	Disability			•	•			
Input	Sensing	•				•	•	
	Audio			•				
	Image				•			•
	Video	•	•					
Output	Recognition	•			•	•		•
	Detection			•			•	
	Control		•					
Analysis	Accuracy		•	•	•			•
	Behavior	•		•	•	•	•	•
	Feedback	•		•	•	•	•	•

pools and are typically conducted in a controlled setting, where the researchers are present. Partially this could be due to the user characteristics of interest; people with disabilities [20, 119], children [151], and students [157]. Another reason could be the challenges in remote data collection as it would require a working prototype [20, 119] or specialized devices from the users [151, 157]. Our teachable object recognition testbed, utilizing a built-in camera in a mobile phone, and existing crowdsourcing platforms allow us to reach a larger participant pool that can be further scaled.

As shown in Table 6.1, the input modality for the teaching set was more often based on sensing [151, 155, 157] and videos [155, 156] with one example for sound [119] and photos [20]. For the last two, participants could not assess the quality of their teaching examples – participants who were deaf/Deaf or hard-of-hearing could not hear the sounds they recorded [119] and blind participants could not see the photos they took [20]. In this paper, we choose images as the input modality for the teaching set. This allows us to tap into a large user group of non-experts that can simply use their mobile phones to take photos in a real-world setting. More so, by choosing an object classification task, an

accessible task to many where they can serve as the oracle, we are given the opportunity to explore how humans teach a high-dimensional decision boundary to machines by feeding them only with few instances. More importantly, this modality allows us to visually inspect the teaching set for common patterns in users' behavior.

Similar to most of the prior work in Table 6.1, our analysis is based on observed behaviors and participant feedback. Leveraging prior work in neuroscience, we examine how non-experts' teaching strategies draw parallels in machine robustness to human robustness, where object recognition involves generalization across size, location, viewpoint, and illumination [158]. While prior work did not include such a fine-grained analysis of the participants' input, it provided insights and anecdotal evidence that guided the design of our studies such as the need for iterations [151, 155, 157], which may vary not only across participants but also due to the underlying algorithm and task [159]. For comparison purposes and time sake, we opted to keep the number of iterations constant at two. Similar to our study, the number of classes was limited (2-5) with an exception of 15 [20], where there were no iterations.

Chapter 7: Understanding Error Identification in Pre-Trained Image Recognition With Blind Users

7.1 Motivation and Introduction

The past few years have yielded vast improvements in image recognition due to advances in machine learning. As computer vision can be used for blind people to access the visual world independently using a camera on their smartphones, many assistive mobile apps (*e.g.*, Seeing AI [124], Envision AI [144]) are deployed to enable them to read text, recognize objects, understand images, and navigate. However, most image recognition systems are built on benchmark datasets with images collected by sighted people, being more likely to have errors with images from blind users.

Prior studies have provided anecdotal evidence indicating that errors in image recognition systems affect blind users' experience with it significantly. As it is hard for most blind users to verify the image recognition results without sighted help, they trust and rely on the output of the recognition system which may not be error-free. For example, MacLeod *et al.* [33] showed that blind users may overtrust the output of an automatic caption generator even when it makes little sense. Image recognition errors would cause more critical problems when blind people interact with real-world objects through their

cameras and computer vision. Jafri *et al.* [131] have warned that such misrecognitions may cause a safety threat when computer vision applications are used for controlling household objects such as stoves and microwaves. Errors are especially non-acceptable when they can adversely affect blind users' interactions with others in a way that may affect how others perceive them. In a recent study, Lee *et al.* [160] discussed how blind users prefer not to get a prediction of passersby gender as potential errors can lead to an embarrassing situation. This echoes some of our previous findings in Chapter 5 where blind participants worried that errors in their dictation could affect others' perception of their intellect.

Similar to Part I, we start our exploration by better understanding and quantifying blind users' ability to identify image recognition errors on their input, an object to be recognized. In this chapter, we focus on blind people's experience with image recognition systems and their errors. We conducted a controlled experiment, which due to COVID-19 had to be simulated in people's homes. The experiment mirrors the controlled lab methods employed in the study in Chapter 5 that included a semi-structured interview and speech dictation task. Our research questions also mirrored those in the speech study. With the semi-structured interview, we aim to get some context on the use of image recognition by this user group and prior strategies for identifying errors. Then, through an error identification task, we explore how well blind users can identify the errors and what strategies they employ.

7.2 Method

To understand blind people's experience with camera-based assistive tools, we conducted a two-part user study including a semi-structured interview and a task of identifying errors of a general object recognizer.

7.2.1 Participants

We recruited 12 blind participants (6 female, 6 male) from campus email lists and local organizations (Table 9.2). The participants ranged in age from 32 to 70 ($M = 54.3, SD = 15.2$). The participants reported being totally blind ($N = 3$), having some light perception ($N = 5$), or being legally blind ($N = 4$). All participants have used smartphones several times a day. While P1 and P2 had some auditory processing disorder and a problem in hearing the high sound, respectively, all participants did not have problems in communication or using a screen reader. All participants reported that they take a photo or record a video at least once a month. When asked to report their levels of familiarity with machine learning in 4-scales: not familiar at all (have never heard of machine learning); slightly familiar (have heard of it but don't know what it does); somewhat familiar (have a broad understanding of what it is and what it does); extremely familiar (have extensive knowledge on machine learning), two participants selected not familiar at all, eight selected somewhat familiar, and two reported being somewhat familiar.

ID	Age	Gender	Level of vision	Age of onset	Familiarity with ML*
P1	39	Female	Light perception	Birth	Not familiar at all
P2	67	Male	Legally blind	55	Slightly familiar
P3	62	Female	Totally blind	Birth	Somewhat familiar
P4	32	Male	Legally blind	20	Slightly familiar
P5	66	Male	Light perception	46	Slightly familiar
P6	61	Male	Light perception	41	Somewhat familiar
P7	70	Male	Legally blind	Birth	Slightly familiar
P8	50	Female	Legally blind	45	Slightly familiar
P9	69	Female	Totally blind	55	Not familiar at all
P10	66	Female	Light perception	Birth	Slightly familiar
P11	33	Female	Light perception	Birth	Slightly familiar
P12	36	Male	Totally blind	Birth	Slightly familiar

*ML: Machine learning

Table 7.1: Participants' characteristics.

7.2.2 Procedure

The user study was conducted in two days. We conducted a semi-structured interview on the first day with questions regarding demographic information, photo-taking experience, and experience with camera-based assistive tools. On the second day, participants completed a task of identifying errors of a general object recognizer.

Semi-structured interview. The interview lasted for one hour. The interview was conducted through an online meeting application, Zoom. The whole interview procedure was video recorded for later analysis. Specifically, participants responded to questions about:

- frequency of using a mobile device, taking photos, reviewing photos, and changing settings of the camera
- purpose of taking a photo, subjects of the photos, applications used to take photos, devices to take photos, the confidence of taking a good photo

- frequency of use, usefulness, and devices for a camera-based assistive tool
- frequency of verifying the recognition results of a camera-based assistive tool, encountering errors, the importance of the errors, difficulty of identifying the errors
- strategy of taking photos for a camera-based assistive tool, degree of understanding how a camera-based assistive tool works

For most questions regarding the frequencies, we employed an absolute 7-point scale adopted from Rosen *et al.* [108] (Never, Once a month, Several times a month, Once a week, Several times a week, Once a day, Several times a day). For some questions about relative frequencies (*e.g.*, how often do you encounter misrecognitions when you use [a name of a camera-based assistive tool]?), the frequencies were measured in a relative 6-point scale (never, very rarely, rarely, occasionally, very frequently, always) [109].

Error identification task. Participants completed a task of taking photos of objects with a general object recognizer and identifying recognition errors 1-7 days after the interview. The devices and objects for the user study were delivered to the participants' houses and the instructions were given through Zoom due to COVID-19. For remote communication, participants are given a laptop computer with Zoom on. We also provided a Vuzix Blade smart glasses with a camera that are also connected to Zoom so that we can monitor the participants' views throughout the study and for later data analysis. Participants used iPhone 8 with an object recognition app that we built for this study. At the beginning of the task, the experimenter provided the names of 15 objects (Figure 7.1). In each trial of the task, participants were asked to select one of the objects randomly, take a photo of it, and get a label from the object recognition app. The label was provided

Figure 7.1: Object stimuli: baking soda, caramel coffee, Cheetos, chewy bars, chicken broth, coca-cola, diced tomatoes, diet coke, dill, Fritos, Lacroix apricot, Lacroix mango, Lays, oregano, pike place roast.

through a synthesized speech. After listening to the label, participants reported whether the recognition was correct or not and how certain they were that the recognition was correct (or not correct). After finishing the recognition trials with the 15 objects, participants went through the objects once more in random order, completing 30 trials in total. Participants were encouraged to think out loud throughout the task. At the end of the task, we asked questions about the difficulty and strategy of identifying errors.

7.2.3 Object Stimuli

For the task of identifying errors from object recognizers, we used 15 objects (Figure 7.1) used in a prior study on examining the blind users' interaction with a teachable object recognizer by Hernisa *et al.* [20]. We followed their approach where the objects are selected to include different shapes, sizes, materials, and visual similarities. The logos or images on the container of some products (*i.e.*, baking soda, chicken broth, diced tomatoes, and diet coke) are slightly different from the products in the prior study because their designs have changed. However, the shape, material, and weight that may affect

participants' tactile perception of the objects were the same for all objects.

7.2.4 Testbed

For the task in the user study, we used a general object recognizer fine-tuned on the photos of objects in Figure 7.1. The base model of the general object recognizer is an InceptionV3 [161] model trained on ImageNet [5]. We fine-tuned the base model on a dataset with photos taken by nine blind participants in a prior study where the participants trained a teachable object recognizer [146]. The dataset included 225 photos for each object, having 3375 photos in total. The fine-tuning was conducted with 500 steps of gradient descent and a 0.01 learning rate. During the task, participants used the general object recognizer app on Apple iPhone 8 (Figure 7.2. When a participant touched a "Scan Item" button on the screen, the app sent an image to a server through HTTP and received the results of recognition from the server where the fine-tuned object recognition model predicted the label of the image.

7.2.5 Measures and Data Analysis

The responses in the semi-structured interview and the tasks are video recorded using Zoom. We transcribed the responses to analyze the results. The images and labels from the object recognizer were saved on the server during the task so that we examine if participants identified errors correctly or not.

Figure 7.2: A screenshot of the general object recognizer.

7.2.5.1 Semi-Structured Interview.

We used a thematic coding approach to find the major themes in the participants' responses [113]. To reduce the subjectivity, two researchers cooperated to code the responses. One of the researchers transcribed the responses. With the transcribed data, the two researchers coded the responses and created initial codebooks. They compared the two codebooks and coded data to resolve the disagreements through consensus. There were 35 disagreements out of 373 answers. After resolving the disagreements, they established a codebook and coded data. In the final codebook, the responses of 17 open questions in the semi-structured interviews included 153 codes.

7.2.5.2 Error Identification Task.

The blind participants' ability to identify the object recognition errors was measured using precision and recall, which are commonly used to measure the performance of machine learning models. Specifically, precision indicates how often the recognition results are actually errors when participants thought the results were incorrect. Recall denotes the proportion of errors that participants correctly identified. We also measured the error rate of the object recognizer using precision, recall, and accuracy.

7.3 Results

The semi-structured interview provides insights into blind people's experience in taking photos and recording videos. The error identification task revealed blind users' patterns in identifying object recognition errors.

7.3.1 Insights from Semi-Structured Interview

The main themes in the questions of the interview included experience in taking photos (or recording videos) and interacting with camera-based assistive apps. We focused on how blind people check the quality of their photos, why they take photos, and how they identify misrecognitions from the camera-based assistive apps.

7.3.1.1 Experience in Taking Photos or Videos

All participants have taken photos at least once a month as shown in Figure 7.3. When they take a photo or record a video, they rarely changed the setting of their cameras or environments. The majority of the participants said they have never changed the settings ($N = 8$). When the four participants changed their settings, they tried to find a place with maximum light ($N = 3$), change the flash setting ($N = 1$), and tried different camera angles ($N = 1$). For example, P8 thought taking photos at home can get the maximum light, saying *"when I'm home, I feel it gives me the maximum amount of light and I get the best pictures. [...] I might move it around a couple of times so that it'll describe it in the most detailed way."* When asked how often they check if their photos are good, many participants responded they occasionally did relative to the frequency of taking photos. The majority of the participants checked their photos several times a month or less ($N = 8$). The participants with low vision reviewed photos with their vision ($N = 4$). They also used automatically generated image descriptions from assistive tools such as Seeing AI and built-in image captioning function in iOS ($N = 5$). For example, P12 who inferred the quality of the photo based on the text recognition results said *"what's relevant are the OCR results I get from it. Especially if there is a garbled section that doesn't fall into a normal OCR error pattern, then I know the photos not good."* The participants also got help from sighted people around them ($N = 3$) or remotely using assistive apps such as Aira [145] and BeMyEyes [162] ($N = 1$).

When asked what they captured in their photos, the most common response was that they capture documents for text recognition ($N = 10$), people ($N = 9$), and objects

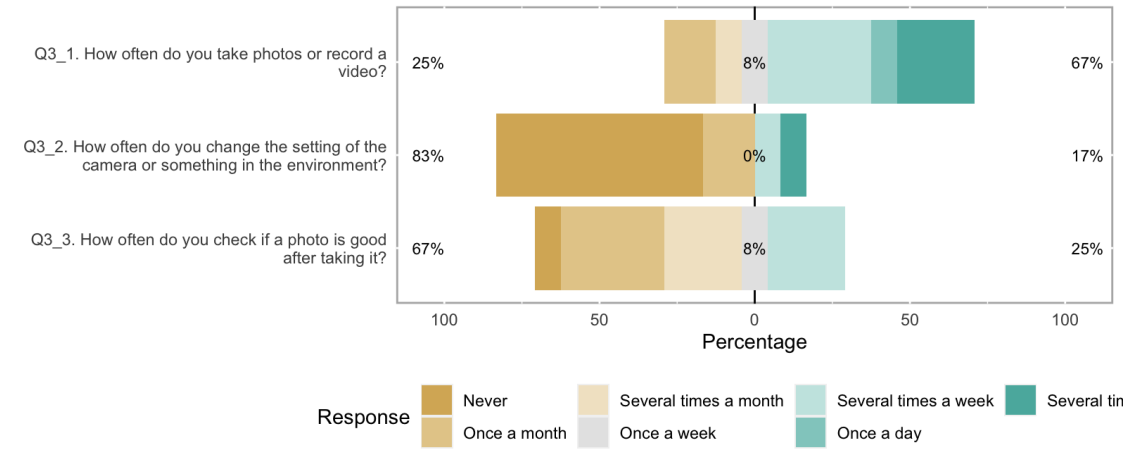

Figure 7.3: Participant responses to questions about their experience in taking photos.

($N = 8$) as shown in Figure 7.4. Similarly, the most common purposes of taking photos or recording videos were for text recognition ($N = 10$), video calls ($N = 8$), and object recognition ($N = 5$). These responses are somewhat different from the findings in a prior study conducted by Jayant *et al.* [142] in 2011 that blind people mostly take photos to capture friends/family for fun while their most desired use for a camera was text recognition. One of the possible reasons for this difference would be that computer vision-based assistive apps became more common among blind people as they have been improved with the advance of machine learning. However, many participants still thought image framing (*i.e.*, centering the object and adjusting the distance between a camera and object) is challenging ($N = 9$). For example, P1 and P5 said *"Making sure the information I'm trying to capture is in the frame of the camera."* and *"I don't know how far away from the object to hold the phone."* Participants also mentioned other challenges: having the focus on the object ($N = 2$), holding a camera steadily ($N = 2$), adjusting the light condition ($N = 2$), finding the right orientation of the object ($N = 2$).

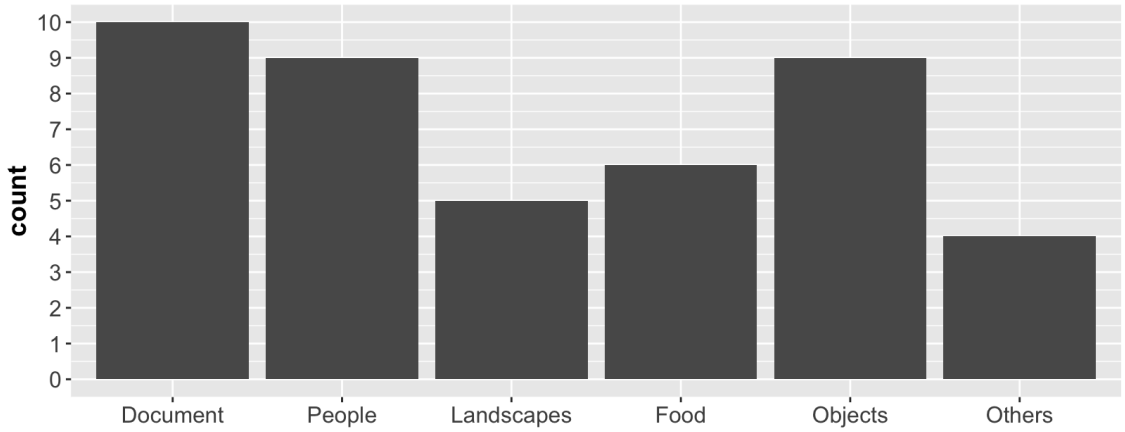

Figure 7.4: What participants captured in their photos.

7.3.1.2 Experience in Using Camera-Based Assistive Apps

We asked participants what camera-based assistive apps they have used regularly. Participants have used eight camera-based assistive apps. We asked questions about their experience in using each app. There were 20 cases (*i.e.*, participant-app pairs). The majority of them have used Seeing AI ($N = 9$) as shown in Figure 7.5. They use other apps with text and object recognition functions such as Google Lookout, KNFB Reader, Super Lidar, Supersense, and Voice Dream Scanner. They used Aira and Be My Eyes to get sighted help remotely through the apps. The participants used the app several times a day ($N = 5$), several times a week ($N = 7$), several times a month ($N = 5$), and once a month ($N = 3$). When asked how frequently they encountered misrecognitions or mistakes from the apps, the participants reported that they rarely ($N = 10$), very rarely, or never encountered such cases. When asked how frequently they encounter errors in the absolute frequency scale, participants found them less frequently than once a week in most cases ($N = 19$). However, one thing to note is that participants would

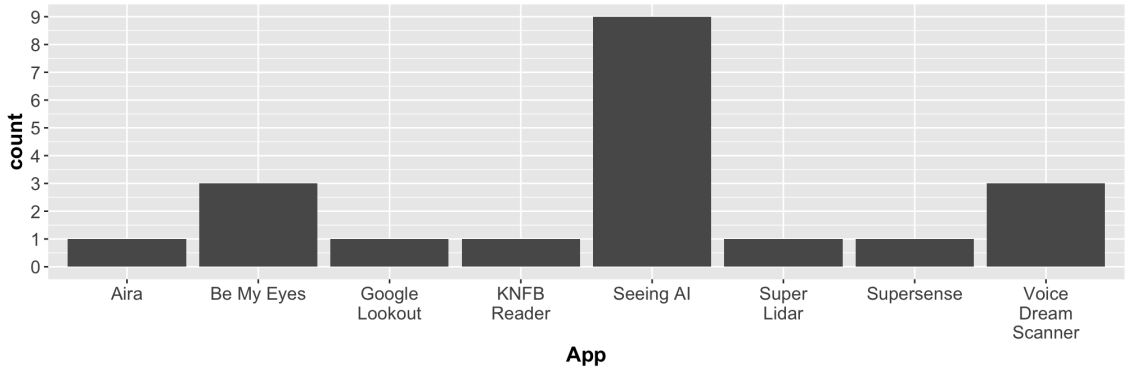

Figure 7.5: Camera-based assistive apps the participants have used regularly.

have not perceived many errors while using the apps without validation considering that participants missed more than half of the errors in the error identification task in the section 7.3.2. Therefore, the frequency reported by the participants would be lower than the actual frequency of errors. To take good photos for these apps, participants tried to find a proper distance between the camera and object ($N = 9$), adjust the orientation of the object ($N = 7$), center the object in the camera frame ($N = 7$) in most cases. This would probably be because most participants thought image framing was challenging as mentioned earlier. When computer-generated feedback for blind photography is available, participants also used them to take good photos ($N = 8$). For example, P12 who have used Voice Dream Scanner said *"It has this system where the louder and steadier the audio tone is, the better you are. There's a certain tone. You've got the perfect picture and you snap it."*

We also asked them how frequently they validate the predictions from the apps (Figure 7.6). The participants have never verified the outputs in most cases while using the apps ($N = 9$). Most of these participants mentioned that they just believe the outputs from the apps ($N = 7$). For example, P2 and P8 said *"if it says it's a $5 bill, I believe*

it" and *"I assume it's correct when it reads it to me."* This response is consistent with a finding from a prior study that blind users usually overtrust computer-vision systems [33]. Some participants did not validate the outputs from the apps because they thought it was easy to find the errors ($N = 6$). When they use a text recognizer, they could identify errors if the outputs do not make sense. For example, P11 who reported that she had never verified the outputs from Seeing AI and Voice Dream Scanner said *"If it tells me a certain thing, I'll know that it actually meant certain numbers. The errors that are sometimes made, they kind of have patterns if you know what it is."* When recognizing objects, they compare the outputs from the apps with what they expected based on the textures, shapes, weights of the object. For example, P6 who never validated outputs from Seeing AI said *"[...] I could say sometimes it does get the canned soup name wrong, but I guess I don't consider it wrong enough to call it wrong."* With some apps, they verified the outputs occasionally ($N = 5$), rarely ($N = 3$), and very rarely ($N = 1$). The most common reasons for verifying the results were that they were unsure with a single output (*i.e.*, needed multiple trials to make a decision) ($N = 8$). For example, P3 said, *"if I'm consistently not getting a result with Seeing AI, then I'll see if KNFB Reader will give me results."*

In most cases, participants agreed ($N = 13$) or strongly agreed ($N = 3$) that they cared about the misrecognitions from the apps as shown in Figure 7.7. They sometimes did not care about the errors because they could understand the outputs from the apps even with some errors (*e.g.*, the errors in text recognition did not change the meaning of the texts significantly) or they did not use the apps for sensitive or important tasks. P8 said, *"It's not the most important thing, because I'm not using it for something critical."*.

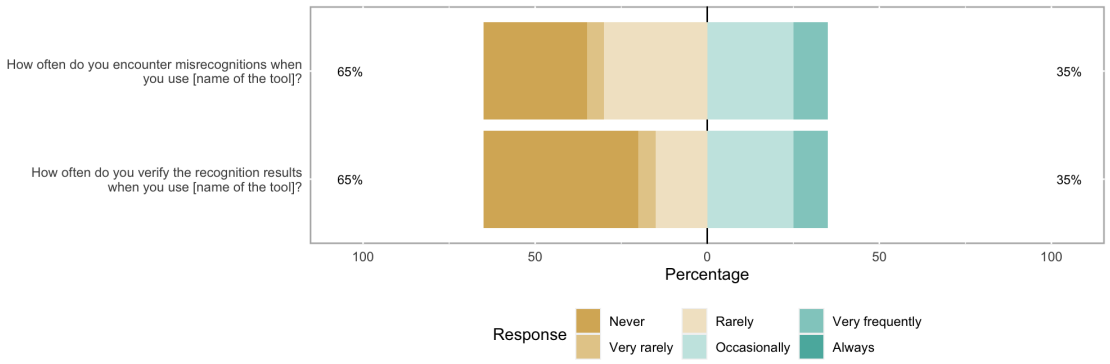

Figure 7.6: Participants responses to two questions about the frequency of encountering errors and verifying the outputs from the apps.

When we asked if there is any situation that they care about the errors more than others, participants' responses were mostly about text recognition and the importance of the contents in texts. The most common responses were when reading bills, currency, expiration date, or other important numbers ($N = 11$). P1 who used Be My Eyes said *"if they don't see the expiration date properly on something and it's expired, you know, I could get sick."* Other situations include reading directions for some tasks ($N = 5$) and reading important documents ($N = 5$). P9 provided some examples of the important documents, saying *"probably when it's something that is connected to legal documents, financial statements, legal financial statements."* The participants' responses were divided on the difficulty of identifying misrecognitions from the apps. Participants disagreed or strongly disagreed that it was challenging when they could easily find the errors using the contexts such as the contents around misrecognized texts and the texture of misrecognized objects ($N = 10$). For example, P1 said *"if they're wrong, I know they're wrong."* P12 said *"I can catch the errors as they come up because often, it's not wrong enough for me to not be able to figure out what it says.* In other cases, participants thought the errors are not clearly

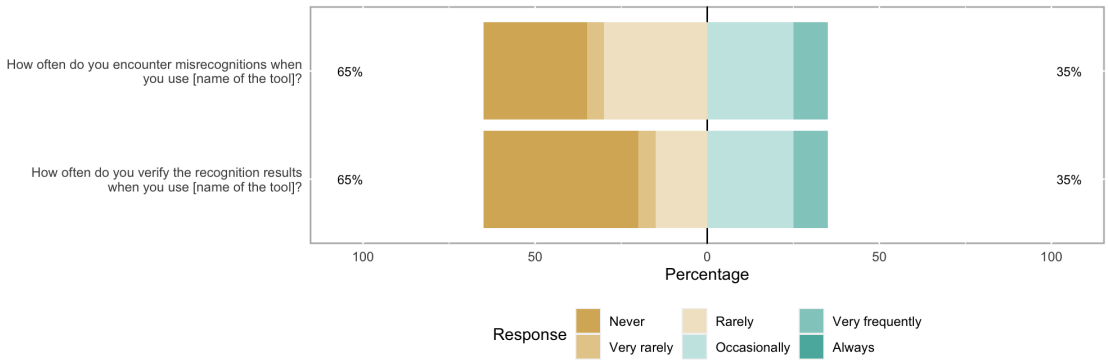

Figure 7.7: Participants responses to two questions about handling errors in the apps.

distinguishable and hard to identify ($N = 9$). P8 who was aware of the possibility of missing the errors said *"If it's wrong, I wouldn't know... I don't even know whether it's wrong or true."* P9 had experiences that sighted people found errors from Voice Dream Scanner when she missed them. She said *"There have been occasions when I didn't detect anything and a sighted person may have indicated there was something that I just did not get."* When we asked participants how they identify the errors, we found that they identified the misrecognitions for themselves based on the contexts (*i.e.*, texts around text recognition errors, textures of the objects) in most cases ($N = 10$). For example, P1 who recognized texts with Seeing AI said *"If the information reading isn't very clear, if I can tell that it's only reading a part of something then I have to readjust it."* P6 who identified objects with Seeing AI said *"[...] if I get a soup, and it's not pronouncing the type of soup, that type of thing [...]."* In other cases, they asked sighted people to clarify ($N = 5$), verify the outputs from the app with multiple trials ($N = 5$).

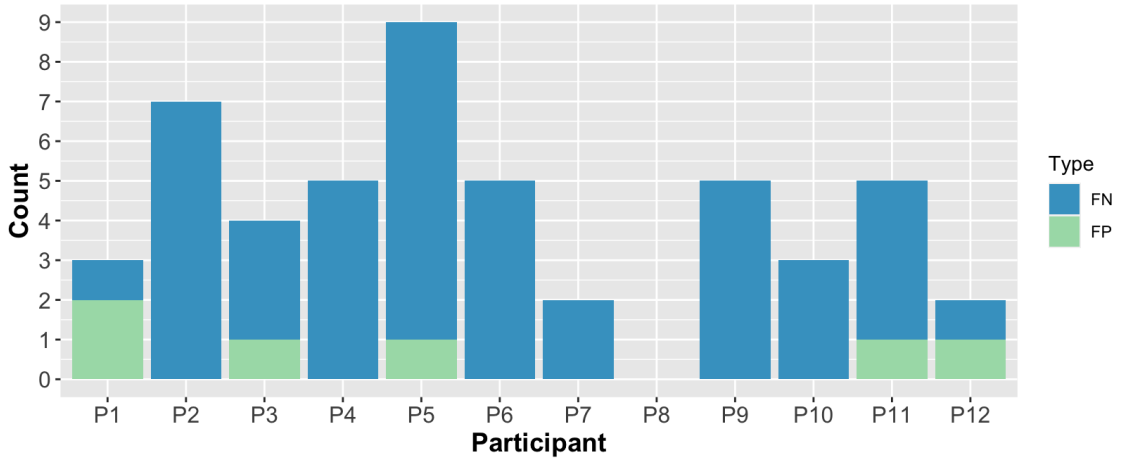

Figure 7.8: The number of missed errors (false negatives, FN) and correct predictions considered as misrecognitions (false positives, FP).

7.3.2 Results from Error Identification Task

Over the 30 trials, the accuracy of the object recognition was 0.76 ($SD = 0.10$) on average. Participants got between 4 and 13 errors during the task. We counted the number of missed errors (false negatives) and the correct predictions considered as errors (false positives). The number of false negatives and false negatives were 3.67 ($SD = 2.46$) and 0.5 ($SD = 6.74$) on average. The proportion of errors that are identified by the participants was 0.49 ($SD = 0.32$) on average. These results indicate that participants tended to believe the predictions from the object recognizer rather than having doubts on them, missing more than half of the errors.

While participants missed many errors, they mostly thought identifying the errors is not challenging. When asked if it was challenging, the majority of them disagreed ($N = 5$) and strongly disagreed ($N = 3$). P8 who has low vision could tell the correct and incorrect predictions based on their vision and the textures of the object. Some participants identified errors by comparing the predictions in multiple trials. For example,

P10 elaborated *"I didn't recognize a mistake until the second similar object appeared. So like the two cans of the Lacroix apricot and Lacroix mango, one of them was incorrect because it was telling me apricot both times."* The errors were sometimes clear for the participants because the predicted and true objects had different textures, shapes, or weights as mentioned by P12. He said *"[...] for example, the diced tomatoes versus the chicken broth, chicken broth is more liquid. It was easy to identify that it was wrong."* On the other hand, only three participants agreed. On the other hand, some participants agreed that it was challenging to identify the errors ($N = 3$). Two of the three participants mentioned that the recognition results were not consistent with an object, making it hard to decide whether the results are correct or not. A participant mentioned that it was difficult to remember all objects explained at the beginning of the study and it made it hard to decide whether the recognition results were correct or not.

7.4 Discussion

7.4.1 Understanding the Contexts of Using the Object Recognition Apps

Looked into blind people's experience with camera-based assistive apps, we observed that the majority of them were about using text recognition apps. Comparing the results in this study and a study in a prior study by [142] in 2011, we could see that both text and object recognition are more popular now than a decade ago. However, the object recognition apps were found not to be as popular as text recognition apps among the participants in this study. One of the possible reasons would be that text recognition would be more reliable than object recognition in practice because the texts in documents

are more structured than the shapes and textures of arbitrary objects. A problem in general object recognition apps is that the recognition results are not fine-grained due to the technical limitations in computer vision [20].

Though both text and object recognition use cameras, the blind users' interactions with these recognition systems would be significantly different. For example, this study showed that some participants identified an error based on the texts around the error while using text recognition and some participants depended on background knowledge of objects such as textures, shapes, and weights to identify object recognition errors. Therefore, we need a more in-depth study that separates the user experiences of text and object recognition apps. Understanding the contexts of using such recognition apps would provide insights for enabling blind users to identify errors as we found that one of the strategies for identifying errors in the error identification task was using the information in context (*i.e.*, recognition results of objects with similar textures). In the error identification task of this study, we assumed that participants would know the candidate objects (*i.e.*, 15 objects in the task) before using an object recognizer. However, the contexts in the task would be different from real use-cases. In reality, for example, the accuracy and strategy of identifying errors would depend on many factors such as whether they know about the objects and the performance of the object recognizer in advance. To understand the impact of the context, we need an in-depth analysis of blind users' experience in using object recognizers in real scenarios.

7.4.2 Interface with Feedback for Identifying Errors

As it is challenging for blind users to aim the object properly, adjust a light condition, check if the background is cluttered, some feedback for them to take good photos would be helpful. Some participants in this study also have used the feedback from camera-based assistive apps to capture a target object. For example, P12 who have used Voice Dream Scanner with feedback mentioned *"It has a system where the louder and steadier the audio tone is, the better you are. When there's a certain tone, you've got the perfect picture and you snap it."* Enabling blind users to take good photos not only results in lower error rates but also affects the trust of the system and the users' accuracy of identifying errors. While prior studies have developed systems that provide feedback for centering an object and adjusting the distance [141, 146], they have not investigated the impact of such feedback on the error rate and blind users' interaction with the errors.

Through the interview with participants, we found that the significance of errors and the frequency of verifying the outputs from recognition apps are affected by the performance of the apps estimated by the participants through past experience. However, as a prior study showed that blind users overtrust automatically generated captions [33], participant's estimation would not be accurate sometimes. Therefore, informing blind users of the certainty of the predictions and the exact performance of the apps would help them identify errors. While prior studies in Explainable AI (XAI) have shown that explaining certainty and rationale behind the output of a machine learning model can improve the trust and usability of machine learning systems [163, 164]. However, many of them are based on visual information inaccessible to blind users or not assessed with

blind people. In future studies, we need to evaluate the impact of feedback with the certainty of predictions and the performance of an object recognition system on blind users' experience.

7.5 Conclusion

In this chapter, we investigated blind users' experience with camera-based assistive apps through a semi-structured interview and measured their accuracy of identifying object recognition errors through a error identification task. Through the interview, we found that participants were divided on the difficulty of identifying misrecognitions of camera-based assistive apps. As they believe the outputs from the apps or some errors are easily identifiable based on the context, they rarely verified the outputs of the apps. However, through the error identification task, we observed that the participants miss more than half of the object recognition errors. Analyzing the participants' strategies for identifying object recognition errors, they use their knowledge of the objects such as textures and weights as well as recognition results of other objects to infer the correctness of the recognition results. The findings emphasize the demand of understanding the context of using object recognizers for blind people and designing interfaces that provide feedback for blind photography and explainable outputs.

Chapter 8: Exploring Error Understanding and Avoidance in Teachable Image Recognition With Sighted Users

8.1 Motivation and Introduction

The previous chapters (Chapter 4, 5, 7) covered both speech and image recognition systems where the machine learning models have been pre-trained and deployed by experts. This and the following chapters switch the contexts from the interactions with the recognition systems pre-trained by experts to the systems that can be personalized by end-users through machine teaching.

As machine learning and artificial intelligence become more present in everyday applications, so do efforts to better capture, understand, and imagine this coexistence. Experts from diverse disciplines are working together and critically examining the impact of algorithmic decisions, their assumptions, and their biases [34, 165, 166, 167, 168]. Error-prone, computationally complex, and failing in ways unexpected by humans, such algorithms called early on for transparency, interpretability, accountability, and control [169, 170, 171, 172, 173]. More recently, these efforts have redoubled (surveyed in [94, 174]), fueled by funding and legal initiatives such as the DARPA Explainable Artificial Intelligence [175] and the European Union's General Data Protection Regulation [176], while feeding into

Figure 8.1: Given an object category, MTurkers are called to choose three object instances and train a *robust* personal object recognizer using their mobile camera. Here we include examples from some of the participants' selected objects.

future initiatives such as the Algorithmic Accountability Act [177].

Machine teaching [4, 178] lies at the core of these efforts as it enables end-users and domain experts with no machine learning expertise to innovate and build AI-infused[1] systems. Beyond helping to democratize machine learning, it offers an opportunity for a deeper understanding of how people perceive and interacts with such systems to inform the design of future interfaces and algorithms [180] – a perspective this paper shares.

Within this paradigm, teachable interfaces [150, 181] explore applications where users can explicitly train a model with their generated data and labels. While facilitating user control, the effectiveness of these applications can be hindered by the lack of expertise or misconceptions about machine learning. Though personalization is often the ultimate goal (*e.g.* [20]), the interactive nature of these interfaces can help users in return to

[1] A term in Amershi *et al.*, 2019 [179] for "systems that have features harnessing AI capabilities that are directly exposed to the end-user."

uncover basic machine learning concepts (*e.g.* [151]).

In this chapter, we examine how people conceptualize, experience, and reflect on their engagement with machine teaching in the context of a supervised image classification task, a task where humans are extremely good compared to machines, especially when they possess prior knowledge of the image classes. As the study in Chapter 4, we reached out to a larger user pool of sighted participants through crowdsourcing. Using a teachable interface for object recognition, we recruit participants ($N = 100$) through Amazon Mechanical Turk[2] to choose three objects in their environment and train a model to distinguish between them in real-time using the camera on their mobile phones, as shown in Figure 8.1.

We build a web-based testbed for a mobile teachable object recognizer and ask participants to train and evaluate it on three objects of choice within an object category (Figure 8.1). Categories represent daily objects that span different characteristics such as size, shape, color, material, and function. Through a performance-based payment scheme [182], participants are called to iterate and reflect over their efforts with the goal of making their recognition models more *robust*. Serving as an oracle, they are tasked with delivering teaching set to the recognition model to help it learn the classification task.

We conduct a contextualized quantitative analysis on the participants' photos, their written responses, as well as their model performance. We find that diversity, important in machine learning, is deemed important by a majority of participants and incorporated in teaching strategies, drawing from parallels to how humans generalize across object size,

[2]https://www.mturk.com/

viewpoint, location, and illumination [158]. Many misconceptions relate to consistency; few think that it is good to be consistent and teach with almost identical examples; others failed to be consistent in incorporating diversity across classes. While participants have good intuition on the importance of discriminatory features in teaching but on evaluating their models, we observe susceptibility to missing edge cases. Last, we see that the majority of participants do not change strategies on a second attempt even though possess a reasonable intuition on what would be important. We see how our findings and insights can help better understand non-experts' interactions with machine teaching and guide the design of future teachable interfaces that can anticipate users' misconceptions and assumptions.

8.2 Method

We deploy our testbed in Amazon Mechanical Turk (IRB #1255427-1) and investigate how non-experts crowdworkers teach a machine a high-dimensional decision boundary such as a fine-grained image classification with a few examples only.

8.2.1 Testbed: Teachable Object Recognizer

To explore how non-experts conceptualize, experience, and reflect on their engagement with machine teaching, we build a web-based teachable object recognizer for mobile phones. Participants can train, test, and re-train it to distinguish between three objects of their choice. In this case, a test corresponds to a 'direct' evaluation [155], where participants take photos of their objects in real-time and observe the model's behavior.

To help us better contextualize our observations, participants also provide background information and feedback[3].

Our machine teaching problem. As shown in Figure 6.1, we adopt Zhu *et al.* [4] machine teaching problem space to characterize the teachable interface in our testbed as a system where the human is the teacher and the machine is the student. The teacher provides, in batch mode, a finite pool of examples consisting of labeled photos of objects as the teaching signal. The teacher takes a model-free approach, treating the student as a black box, though we anticipate that humans may already have some assumptions on how the black box works or should work. The student, employing a convolutional neural network, does not anticipate teaching, *i.e.*assuming training examples are independent and identically distributed and that there are no errors. More so, the teacher is considered a friend, *i.e.*no adversarial training. Last, we assume that the teacher uses heuristic teaching methods to improve the performance of the student, the object recognition model in our case. We aim to better understand these heuristic methods, factors they may relate to, as well as assumptions that people may have.

Model. For each user, our testbed creates a new convolutional neural network using the Google Inception V3 [161] pre-trained on ImageNet [5]. Every time the user provides a teaching set, the last layer of the pre-trained model gets replaced with a new softmax layer and re-trained with the user's images with 500 steps and a gradient descent learning rate of 10^{-2}. The models are trained on our 8 GPU server in real-time asynchronously; the app continues to run and ask users for open-ended feedback while the training continues in the back. The web interface communicates with the server using the Flask API [183].

[3]Questions and prompts can be found in the supplementary material.

Interface. As shown in Figure 8.2, initially the testbed asks for background information, technology experience, and familiarity with machine learning. Then, it provides five object category options: bottle, cereal, drink, snack, and spice, with three sample icons for each category indicative of the preferred shape. Categories are inspired by prior work on personal object recognizers [20] and are engineered to elicit objects that are present in daily life but differ in size, shape, color, material, and function. Participants can choose to train only on one of the categories. To avoid object shape or size from being a factor in any observed inconsistencies between the classes, they are asked to use objects (a total of three) that fall within the same category; three, the smallest number for multiclass classification and previously used in teachable interfaces for non-experts [181], minimizes challenges in finding different object instances within a category in a real-world environment as well as the task completion time (already 40 mins long). After labeling their objects, participants are guided through five interactions with the machine learning model (the student)[4]:

Preliminary test (TS0): Participants are asked to take photos of their objects to see if the existing non-personalized model can recognize them. The instruction reads: *"Take a photo of an object (name at the top) by tapping on the camera screen. The existing model will try to predict it."* Given an object label displayed at the top, one takes a photo of the corresponding object and sees the recognition result (a label displayed for 3 seconds). This repeats 15 times (5 times per object in random order). As expected, during this interaction recognition results will not match participant's labels as the generic model is based on Google's Inception V3 and is not yet personalized. There is a dual motivation

[4] All instructions can be found in the supplementary material.

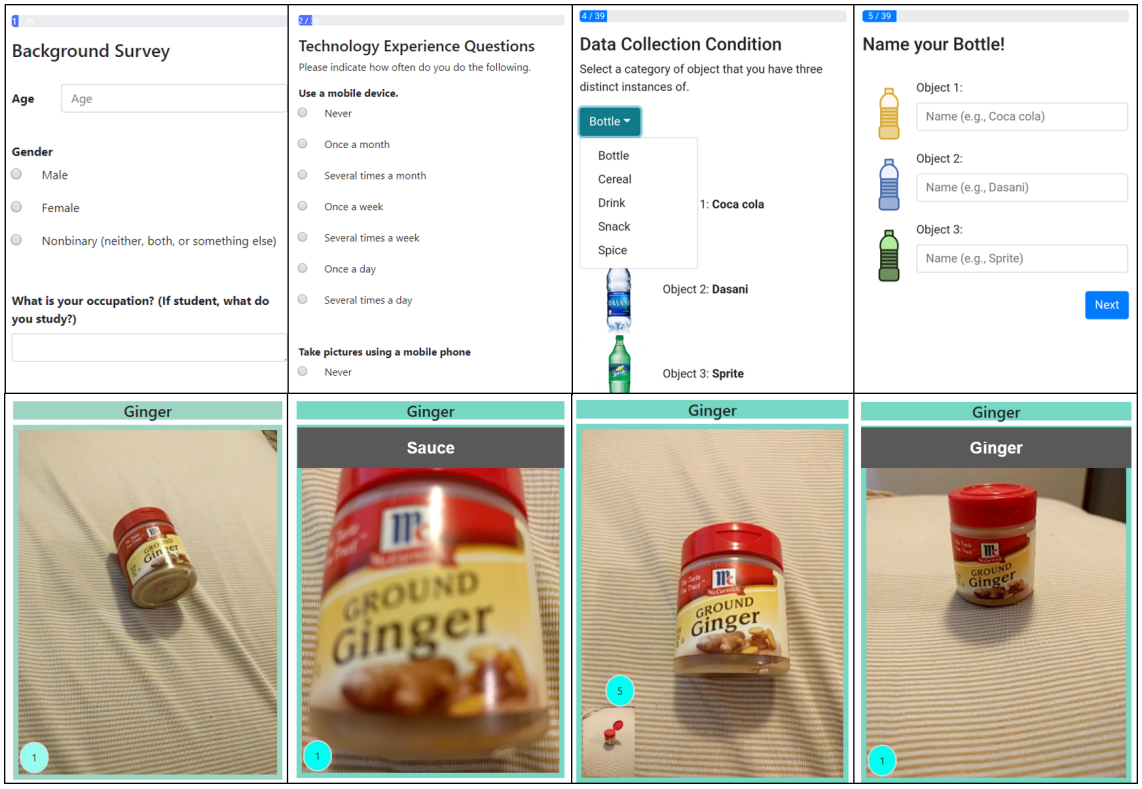

Figure 8.2: Testbed screenshots: questionnaires, category selection, object labeling, and camera view in training and testing.

behind this interaction. First, it helps familiarize with the interface, which simulates the native camera app. Second, it helps collect evaluation examples unbiased from one's teaching experience that is to follow.

Train 1 (TR1): Participants are asked to train the object recognizer with the following instructions: *"Train our object recognizer to identify robustly your objects anywhere, anytime, for anyone. We will randomly choose one of your objects and ask you to take 30 photos of it. You will be paid $2 extra if your examples pass our robustness test."* Here, we hint that model robustness means to be able to recognize an object anywhere, anytime, for anyone. Motivated by Ho *et al.* [182] performance-based payment scheme, we also create the impression of a 'secret' test distinguishing examples best for robustness, though on our end this is merely a naive quality examination (*e.g.*photos of objects in a screen rather than in the real-world). As shown in Figure 8.2, given an object label displayed at the top, participants take 30 sequential photos. This repeats 3 times (1 time per object in random order). Thus, the first teaching set comprises 90 photos (30 per object).

Test 1 (TS1): Similar to TS0, participants are asked to *"Test the trained object recognizer again to see how robust it is."* Here, recognition labels match participants' labels except in cases of misclassification, where an object is misrecognized as one of the other two. Again, no confidence scores are shown.

Train 2 (TR2): Participants are given an opportunity to re-train their model from scratch with the following instructions: *"You told us what you would do differently, now show us! On the next screen, take 30 more pictures of the requested object. You will be paid $3 extra if this training does better than the previous one in our robustness test."*

Test 2 (TS2): As in TS1, users can test the re-trained model. The instruction given

to the participant was *"The object recognizer is trained again. Test the trained object recognizer."*

Eliciting Feedback. The testbed includes the following open-ended questions: *"What did you think was important to consider when training the object recognizer?"* after TR1; *"If you were to retrain the system to make it more robust, what would you do differently?"* after TS1; *"How did you position the object in the image?"*, *"How did you decide the distance of the camera from the object?"*, and *"How did you decide which side of the object is visible in the image?"* at the end.

8.2.2 Participants

We recruited 143 participants over 10 days. However, data from 43 were excluded from the analysis – 7 helped in piloting, 1 used the same object for all classes, 3 took photos of objects in display screens, 2 took photos with no objects. The other 30 had technical problems by attempting the task simultaneously with our system failing to distribute them across the 8 GPUs, losing data from 12, and interrupting the task for the other 18; all were compensated and the bug was fixed. The 100 participants who were included in the dataset ranged from 20 to 60 in age (μ=32.6, σ=8.3); 49 were male, 50 female, and 1 non-binary with 90 reporting being right-handed. No one reported a visual or motor impairment. As shown in Figure 8.3, the majority of participants are frequent users of mobile devices taking photos with them weekly, though many of them don't use any applications for recognizing objects, food, or plants. When asked about familiarity with machine learning, 6 reported never having heard of it, 45 had heard of it but didn't

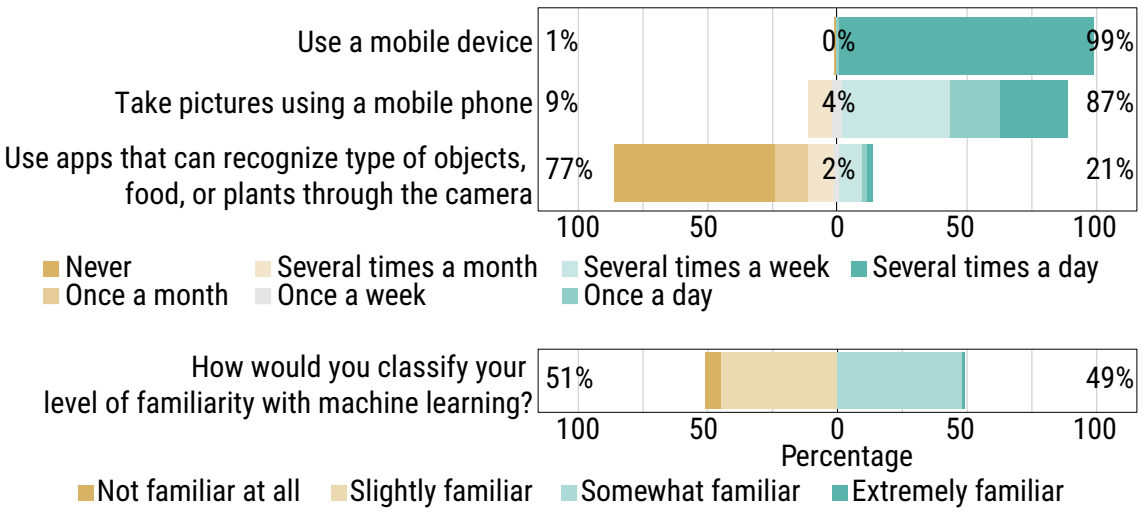

Figure 8.3: Participants' technology experience and familiarity with machine learning mostly ranging from slightly (have heard of it but don't know what it does) to somewhat familiar (I have a broad understanding of what it is and what it does).

know what it does, 48 had a broad understanding of what it is and what it does, and only one reported having extensive knowledge.

8.2.3 Procedure

With the goal of attracting non-experts in machine learning, we opted for a HIT description that minimizes technical terms: *"You will be asked to take photos of everyday products such as soda cans, cereal boxes, and spices to teach your phone to automatically recognize them. To see how well the object recognition works you will test it by giving a single photo at the time."* A warning message was displayed if participants attempted to start the study from a device other than a mobile phone. Only one participation was allowed.

Through piloting, we estimated that a study session could be successfully completed within 30-40 minutes. Adopting a $15/hour compensation rate [184] all participants

received a total of $10 once all the data collection was completed. To incentivize participants, we used a performance-based payment scheme [182], where this amount was split as $5 flat participation, $2 bonus for passing *"our robustness test"* in the first attempt to train, and $3 bonus for achieving a better performance in *"our robustness test"* the second time around. Given that objects differ across participants it was not possible to have an ideal *'secret robustness test'*; the bonus was decided merely on a quality check. While the testbed's connection is persistent and one could do other tasks in between, we observe that participants took on average 35.57 minutes (14.21-79.86, σ=12.85) to complete the study, very close to our estimates.

We explore how participants conceptualize, experience, and reflect on their engagement with machine teaching by looking at the photos they took for the teaching and testing sets as well as changes in their behavior when repeating the process. Observations are contextualized with participants' responses.

Visual Attributes in Photos. We collected a total of $22,500$ photos from 100 participants across all training and testing interactions. To uncover patterns in participants' teaching strategies, photos were coded using thematic coding [113]. Two researchers independently created initial codebooks of visual attributes in photos across four dimensions, *i.e.*size, location, viewpoint, and illumination; prior work on visual object understanding [158] indicates that our ability to recognize objects generalizes across these dimensions. We want to see how participants draw parallels from their understanding of robustness in these dimensions to enable machines to do the same.

Researchers discussed disagreements to produce a final codebook, shown in Tables 8.1–8.3 with examples in Figures 8.4 and 8.5. There are two types of attributes: binary

Table 8.1: Variation attributes, true if a variation is present for at least one object.

Variation	Definition
VSizeDist	True if camera **distance**, ratio of object height to frame, differs for two or more photos using [0, 0.25), [0.25, 0.5), [0.5, 1.0), and [1.0, ∞) bins.
VLocBg	True if the **background** differs for two or more photos, *i.e.*different locations or perspectives of a space.
VViewSide	True if the **side** of objects differs for two or more photos.
VViewAngle	True if the **angle** between the camera and the object with the same side of an object differs for two or more photos.
VViewPos	True if the **position** of the object in the camera frame, center, top left, top right, bottom left, or bottom right, differs for two or more photos.
VIllumExp	True if the **exposure** to light differs for two or more photos taken at the same location.
VIllumSrc	True if the **source** of light differs for two or more photos because they were taken at different locations.

and count. Binary attributes capture the presence of variation or inconsistency within a teaching or testing set of photos. If a participant varied photos for an object along with an attribute such as distance (*VSizeDist*) or background (*VLocBg*), the corresponding attribute is 1; otherwise 0. Similarly, variation inconsistency across the three objects is captured through binary attributes, named *ISize, ILoc, IView, IIllum*. Count attributes indicate the number of photos within a set with a certain characteristic such as the presence of participant's hand (*CHands*) and use of flashlight (*CFlash*) or a quality issue such as dark (*QDim*) and blurry (*QBlurry*) photos. There was substantial agreement (Cohen's kappa=0.80).

Subjective Feedback. Participants' responses to the open-ended questions were also analyzed with a thematic coding approach [113]. The same two researchers who coded the photos, created initial codebooks and merged them through discussions resolving disagreements. Responses were coded independently with a substantial agreement (Cohen's kappa=0.73).

Table 8.2: Inconsistency attributes, true if there is an inconsistency in variation across the three objects.

Count	Definition
ISize	True if the camera **distance** varies in the photos for one or two objects but not all three.
ILoc	True if the **background** varies in the photos for one or two objects but not all three.
IView	True if size, angle, or position capturing **viewpoint** varies in the training photos for one or two objects but not all three.
IIllum	True if light exposure or source capturing **illumination** varies in the training photos for one or two objects but not all three.

Table 8.3: Count attributes, number of photos with a given characteristic including those looking at quality issues.

Count	Definition
CCrop	Number of photos where the object is **cropped**, *i.e.*object is close to the camera, out of frame, or obscured by another object.
CReshape	Number of photos where the object was **reshaped** (*e.g.*opening a lid of a package).
CContents	Number of photos where the **contents** inside a package was taken out of the container or the inside of the package is visible.
CNoBg	Number of photos where were the background is **not visible** because the photos are filled with the object completely.
CPlainBg	Number of photos where the background includes two or fewer colors with **no or very simple textures**.
CClutBg	Number of photos where the background is **cluttered** with objects other than the object of interest.
CTextBg	Number of photos where the background includes a wall, floor, or furniture with **texture**.
CHands	Number of photos where the participant's **hand(s)** is visible in the photo.
CLogo	Number of photos where the side with the **logo** (or label) of the object was visible in the photos.
CFlash	Number of photos where the brightness varies in different parts of the photo like using **flashlight**.
QSmall	Number of photos where the object is too **small** (height of the object ¡ 25% of the height of the photo).
QDim	Number of photos where the brightness of the photo is too **dark** to recognize texture or edge of the object.
QBlurry	Number of photos where the object of interest is **blurry**.
QIrrelevant	Number of photos where the photo includes only **irrelevant** objects without the object of interest.

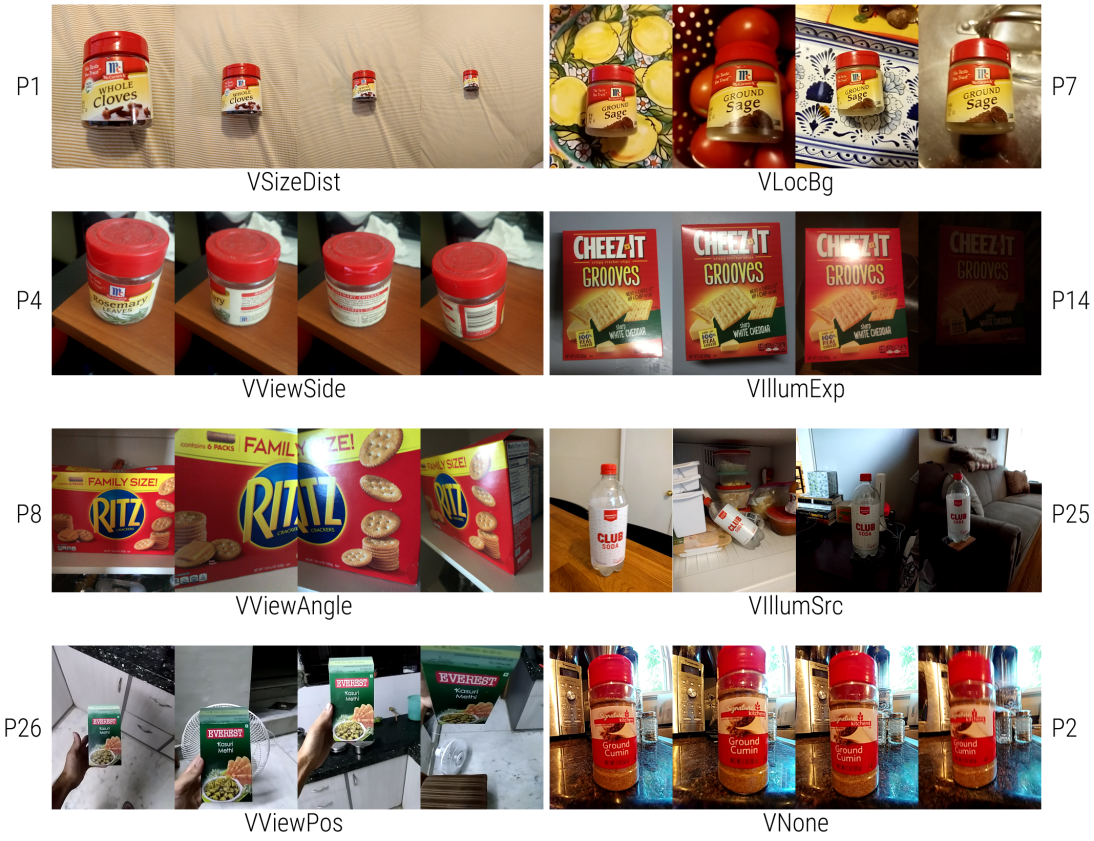

Figure 8.4: Examples of variation attributes in teaching sets.

8.3 Results

8.3.1 Teaching and Debugging Strategies

We explore how variation[5], inconsistency, and other attributes manifest on participants' image sets when they are first called to train the object recognizer on objects of their choice.

Incorporating diversity in teaching. Diversity plays an important role in machine learning [185]. When incorporated in the teaching set, it ensures that examples can provide more discriminatory information to help the model learn. By looking at participants'

[5]A preliminary analysis of this appears in a work-in-progress [1].

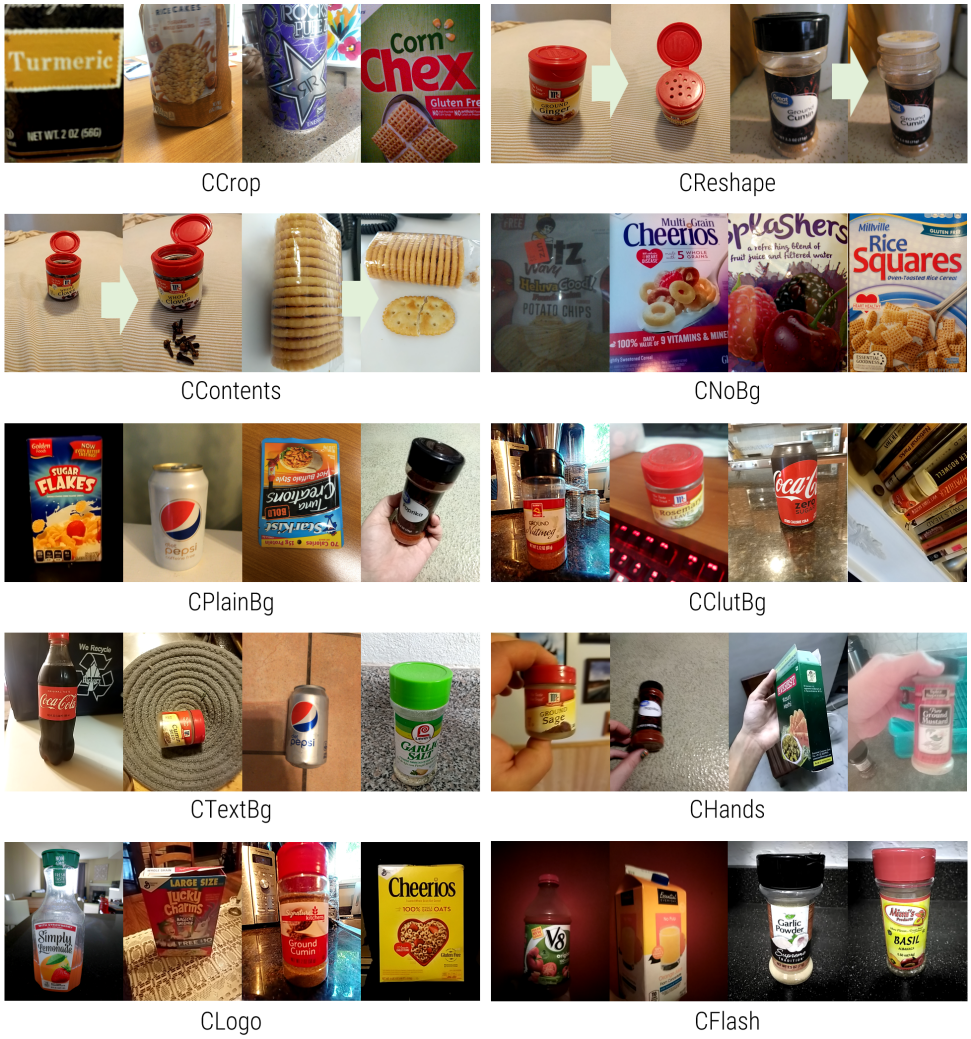

Figure 8.5: Sample photos considered by the count attributes.

photos (results in Figure 8.6) and by reading their responses, we find that the majority of the participants share this intuition, but not all. In detail, 23 participants (age 21–60, μ=37.57, σ=9.87) did not include any kind of variation in their TR1 teaching set – 3 of them reported never having heard of machine learning, 12 had heard of it but did not know what it does, and 8 had a broad understanding of what it is and what it does. Immediately after training, when asked about what they considered important, 5 participants referred to the need for consistency, which in this context contradicts the way

machines and people learn. For instance, P6 said *"I figured I needed to be consistent when I took the picture so they looked similar."* and P30 *"Keeping the pictures the same."* Others, who did not consider this type of consistency, mentioned that it is important to have a good quality photo where the object is well framed (4) with visible labels (8) and images that are clear (6) with ample light (2). Without even having tested their model, P2 said: *"Getting different angles and perspectives so the trainer could recognize it more easily"* – a contradiction to their initial teaching set that had no variation. We observed that in TR2, P2 reflected on this observation and varied both the object size and viewpoint. Only two other participants from this group did so as well, P5 and P18. They said having the "name and color in" is important in TR1 but also varied the camera distance (P5) and angle (P18) in TR2.

However, the majority of participants ($N = 77$) diversified examples in their first attempt. They varied either size ($N = 65$) or viewpoint ($N = 63$), with some considering location ($N = 39$) and illumination ($N = 19$). Light exposure was least diverse ($N = 4$). Looking at responses on important considerations for training, many participants ($N = 52$) mentioned these strategies[6] and reflected on the need for diversity with concrete terms such as *"different", "various", "all", "many", "multiple", "every", "variety", and "difference"* combined with *"angles", "views", "sides", "facets", "background", "lighting", "distance", and "positioning"*. These terms correspond to the four dimensions of our coding scheme informed from prior work on visual object understanding [158], highlighting that humans' strategies for machine teaching parallel their own abilities.

[6] All questions, instructions, and prompts prior to training were carefully edited not to prime participants towards our coding attributes.

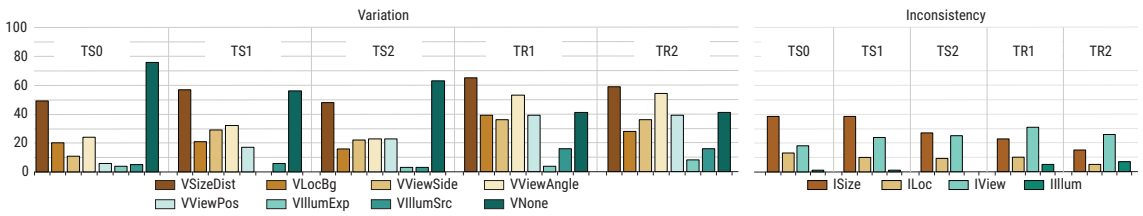

Figure 8.6: Number of participants per variation and inconsistency attribute across all five interactions with the model: preliminary test (TS0), train 1 (TR1), test 1(TS1), train 2 (TR2), and test 2 (TS2). The graphs on the left indicate how participants incorporate diversity in their photos in terms of object size, viewpoint, location, and illumination when they train and debug their models.

However, only 11 participants (age: $\mu = 34, \sigma = 8.71$) incorporated diversity in their teaching set across all four dimensions – 3 reported having heard of machine learning with no further understanding, and 8 had a broad understanding of what it is and what it does.

Being fair and consistent between classes. Model consistency across classes is a desirable trait in machine learning with many social implications for fairness, whose definition is still being debated in the community (*e.g.* [186, 187]). There is anecdotal evidence on non-experts learning to balance class proportions in the training set over multiple iterations [155, 157]. By keeping the number of training examples constant, we look into their behavior across other potential disparate treatments. Given that many participants considered diversity important for good performance, we explore how fair[7] (*i.e.*consistent) they are in incorporating diversity across their three objects, with results shown in Figure 8.6. Beyond the 23 participants who did not introduce any variation for any object, we find that there were 30 other participants that were consistent. This is

[7]In this work classes are object instances that fall within the same category and consequently share similarities such as shape, size, and material in the context of the decision making task of incorporating variation. Thus, we consider "individual fairness" [188], where "similar individuals should be treated similarly", and explore whether object instances within a category are being treated the same by a participant when introducing variation in the training photos.

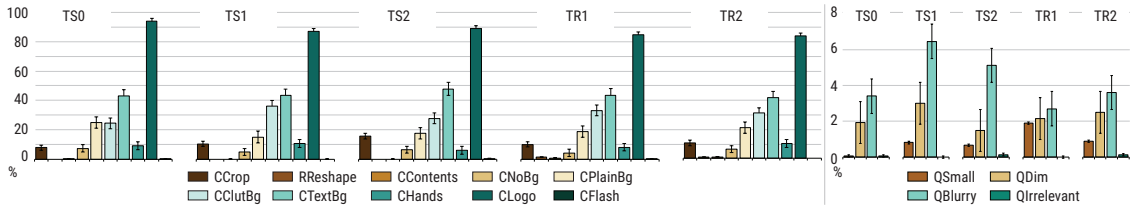

Figure 8.7: Percentage of photos per participant given a count attribute, with standard error as error bars. Participants took photos mostly with the logo on it and many of them against a textured or cluttered background. Often the objects were cropped in the camera frame and sometimes participants' hands were included in the photos. Surprisingly, few participants opened the object and trained the model on their content as well. The most common quality issues were blurry and dim photos though not that prevalent.

promising, especially since this included participants from all levels of familiarity with machine learning: not familiar at all ($N = 1$), slightly familiar ($N = 11$), somewhat familiar ($N = 17$), and the only participant in our study that reported being extremely familiar ($N = 1$). While none of these participants explicitly mentioned consistency as important, we find that more than half of them ($N = 16$) continued doing so in their second attempt at training, in TR2. For the remaining 47 participants, their inconsistencies were found in variations related to all four dimensions: object size ($N = 21$), viewpoint ($N = 31$), location ($N = 10$), and illumination ($N = 5$).

Deciding what to show in the teaching set. We analyze the fine-grained count attributes in teaching and training sets (Figure 8.7) to uncover common teaching patterns across participants. Khan *et al.* [189] observed that one of the most prominent teaching strategies for a binary classification task among non-experts, called the *extreme* strategy, is consistent with the "curriculum learning" principle [190, 191], where participants start with the most extreme examples and continue with those closer to the decision boundary[8]. While our batch teaching task does not allow for a similar sequential analysis, we find

[8]In the Khan *et al.* [189] study participants did not generate the examples but they ordered them as most representative of the two classes and chose to teach one by one using all of them or a subset.

that almost all participants ($N = 98$) included the logo (or label) of objects in their teaching sets; on average 84.9% ($SD = 25.0$) of any participants' images included logos. This indicates that participants understand that logos and labels tend to include the most discriminatory features, which serve as the most extreme examples. Then, through variation, they add less discriminative viewpoints that are closer to the decision boundary. Indeed, 18 participants explicitly mentioned logos or labels as being important in training. For instance, P36 said *"... trying to have a constant label view"* and P46 *"... a clear shot of the front of the package with minimal background interference."* When looking deeper at these responses though, we find that many of the participants assumed that the machine would read the text. For example, P28 said *"It [the model] recognizing the different cereals by name"* and P44 *"Getting a clear shot where the writing and the size are clear."*

In terms of the background, we find that the majority were textured ($N = 66$) or cluttered ($N = 62$), while many used plain ($N = 48$) and a few none at all ($N = 11$) – the latter two are preferred since very few varied the object location. We observe that 26 participants included their hands in the photos. The presence of hands has been leveraged to better distinguish objects by modeling the contextual relationship between grasp types and object attributes [192] or to estimate the object of interest in a clutter environment [140, 146]. However, given this study's fine-grained task, the grasp is expected to be similar across objects of the same category. Thus, the presence of the hand doesn't really help, especially if it is not applied consistently across classes. More surprisingly, we observe that 8 participants reshaped their objects, *e.g.*opened the lid, and 4 decided to train on the content of the object as well, *e.g.*cinnamon powder. When asked what is important for training, one of these participants, P76, said: "Getting lots of

different angles and different ways the spice could be portrayed." In general, there were not many photos with quality issues. Participants took clear photos in most cases and many of them mentioned the importance of image quality in their responses, but some ($N = 36$) mistakenly took a few blurry photos. Also, objects sometimes appeared too small ($N = 17$) and occasionally the light was dim ($N = 9$).

Debugging and including edge cases in testing. When asked to evaluate their model in TS1, many participants ($N = 30$) did not diversify their images at all – 2 of them reported never having heard of machine learning, 17 had heard of it but didn't know what it does, and 11 had a broad understanding of what it is and what it does. This means that they did not check whether the recognizer is robust. We also find that compared to training, fewer participants diversify their testing set across object size ($N = 57$), viewpoint ($N = 49$), location ($N = 21$) and illumination ($N = 6$). This could be explained by many factors such as: a smaller number of photos in testing (15) compared to training (90); difficulty in conceptualizing robustness; assumptions about machine's generalizing capabilities; not anticipating future uses of the model under different circumstances; or simply minimizing efforts for this HIT. Logos were still included by the majority of the participants ($N = 98$) and the same number of participants ($N = 11$) took photos that did not include any background, keeping their testing data consistent with their training examples. Similar to what Zimmermann *et al.* [157] observed, participants "enacted [testing] practices wherein their models appeared to have high reliability but questionable validity." We also find that participants took fewer photos with plain background ($W = 756$, $Z = 2.17$, $p = .030$, $r = 0.15$), and objects that were too small ($W = 126.5$, $Z = 2.61$, $p = .011$, $r = 0.18$) using a Wilcoxon signed-rank test. None of the interesting

object reshaping, or content images present in training, carried over to testing; a similar behavior to Kacorri *et al.* [20], with "exaggerated" variation in training unobserved in testing.

8.3.2 Changes in Teaching Strategies Through Iterations

Prior work indicates that the interactive nature of teachable interfaces can help users uncover machine learning concepts [151]. We ask participants whether they would do something differently were they to retrain the model for a second time and offer a bonus if they could make it even more robust.

Updating teaching strategies to improve performance. "Is this information a signal or noise" was one of the most common debug strategies by experts [193]. We investigate whether participants employ a similar approach by comparing TR2 to TR1 in terms of the variation, inconsistency, and other image characteristics, which serve as information signals for the model. Using a McNemar test for binary and Wilcoxon signed rank test for count attributes, we find the only significant difference is variation of location as observed by changes in the photo background (VLocBg). More participants diversified the background in their teaching set on the first attempt than the second ($\chi^2(1, N = 100) = 4.35, p = .037, \phi = 0.21$, the odds ratio is 11.86). As in Zimmermann *et al.* [157], we suspect that participants were trying to maximize performance by increasing consistency between their training and testing data, even though in our prompts we had defined robustness as ability to recognize the objects anywhere, anytime, for anyone. No other significant differences were observed, though this could be partially explained by

limitations in the binary nature of our variation and inconsistency attributes failing to capture changes in magnitude. We shed light into other possible explanations by looking at participant's responses.

When asked about what they would do differently if they were to retrain, some ($N = 22$) said *"nothing"*, *"wouldn't do it differently"*, and *"would not change anything"*. Few said they had nothing to change because they were satisfied with the performance in TS1 ($N = 6$). For instance, P23 said *"Nothing it seems very robust after the learning phase."* This was not a surprise given that in TS1 participants did not opt for a thorough evaluation, as discussed above. *"Having no idea what to change"* was also mentioned by some ($N = 19$) reflected by terms such as *"not sure"*, *"unsure"*, *"I can't think of anything"*, *"have no idea"*, or *"don't know"*. Indeed, we find that the models of these 22 participants perform well on their own test data with an average F_1 score of 0.981 ($SD = 0.048$)[9] and significantly better than the rest of the participants ($U = 1472, Z = 5.22, p < .001, r = 0.52$); a trend that carries over to the second attempt.

Few participants wanted to change elements of the teaching process such as improving the testbed ($N = 3$), taking photos faster ($N = 1$), adding more classes ($N = 2$), or adding more samples ($N = 6$). Yang *et al.* [193] characterized the latter as "most non-experts' only strategy to improve a model's performance." Others focused on improving the quality of their teaching set such as better focus ($N = 5$), more light ($N = 2$), show labels ($N = 2$), better framing with a certain distance ($N = 1$), and centering ($N = 1$). Few participants ($N = 2$) explicitly mentioned the importance of the background, with P83 saying *"I would try to change the color of the background to ensure that it knows*

[9]Only recognition labels are available in testing and no scores.

what the actual object is. I think it was confused by the curry because of the black stove background which may look like the black cap of the cumin." Surprisingly, one participant (P85) pointed to discriminatory limitations of their objects uncovering challenges in fine-grained classification by stating *"Change objects to not look so similar."*

Last, some participants ($N = 22$) explicitly indicate that adding more variation in their training set is something they would do. For instance, P14: *"I would take a wider variety of angles"* and P21: *"Take picture from many different locations lighting and positions."* Only one, P36 mentioned doing so in testing, *"Test different sizes"*. When examining what they actually did in their second attempt at training, we find differing approaches: some indeed started incorporating new variations ($N = 13$), some perhaps changed the magnitude as variations were present in both first and second attempt ($N = 5$), and others ($N = 4$) did not make those changes. While variation for these 22 participants was mostly limited to the 4 dimensions (size, viewpoint, location, and illumination), few other participants ($N = 5$) indicated that they would also include different forms of the same object, *e.g.*different containers, perhaps difficult within this study.

8.3.3 Analysis of Performance

We report the performance of the models that the participants train by looking at the predicted labels during the first and second round of testing using the F_1 score measure (F-score).

Relating observed behavior to performance. Participants achieved on average a

0.75 ($SD = 0.38$) F-score in their first attempt to train the model. Using a multiple linear regression, we explore how attributes capturing their behavior in teaching and testing may relate to the relative performance of their models. While this performance is far from an ideal controlled robustness[10], it can provide some context for the observations above such as participants' behavior in the second attempt. We use a square root transform of the F-score[11] as the dependent variable. As independent variables, we use variation, inconsistency, and count attributes in TR1 and TS1 and their interaction. For model selection, we use stepwise variable selection based on Akaike information criterion (AIC) [194] with results shown in Table 8.4. We find that only 28% of the variability in recognition performance is accounted by this model, as indicated by the adjusted R-squared metric. While this is modest, it is not surprising, as there are many factors that can contribute to the performance of an image classification algorithm. For instance, performance can vary based on object similarities, a common challenge in fine-grained classification; a similarity that is not directly captured by our attributes.

In training, we find that variation in light exposure (VIllumExp) relates positively with the F-score, though very few participants included this type of diversity in their teaching set. We also see that the number of images where the object is taken against a plain background (CPlainBg) has a negative relationship with model performance. Though counter-intuitive, we suspect that lack of diversity in the background might have contributed to a model that does not generalize well, *e.g.*when tested. This seems to be supported by the negative relationship of the number of cluttered background images during testing.

[10]Such a neutral test is unrealistic in our study since participants choose different objects in different environments.

[11]Transformation is used to meet the normality assumption.

Attempt	Variable	Estimate	Std. Error	t value
TR1	(Intercept)	0.939	0.048	19.79***
	VIllumExp	0.167	0.063	2.64**
	VIllumSrc	-0.076	0.049	-1.55
	CCrop	0.000	0.002	0.12
	CPlainBg	-0.002	0.001	-2.50*
	CTextBg	-0.001	0.001	-1.55
TS1	VSizeDist	-0.068	0.037	-1.81.
	VViewSide	0.108	0.038	2.83**
	VViewPos	-0.089	0.045	-1.97.
	CCrop	0.048	0.012	4.04***
	CClutBg	-0.007	0.003	-2.14*
	QBlurry	-0.016	0.009	-1.74.
TR*TS	CCrop	-0.001	0.000	-3.16**

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.157 on 87 degrees of freedom
Multiple R-squared: 0.3681, Adjusted R-squared: 0.2809
F-statistic: 4.223 on 12 and 87 DF, p-value: 3.195e-05

Table 8.4: Modeling recognition performance based on attributes capturing variation, inconsistency, and other characteristics.

In testing, we find that variation in object size (VViewSide) relates positively with the F-score. We also see that the number of images where objects appear to be cropped (CCrop) has a positive relationship with model performance. A plausible explanation could be that these attributes capture participants' behavior of zooming in on the object's most discriminative features, thus helping the model to distinguish objects. However, when considered as an interaction between training and testing (TR1*TS1-CCrop), this attribute appears to be negatively related to the model performance perhaps pointing to the sensitivity for consistency between the two – if you crop objects in one case, then it helps to do so in the other as well.

Improving performance the second time around. As shown in the previous analysis, we observe few changes in participants' teaching strategies in the second training as captured by our attributes – though some participants said they would do things differently.

We find that this is also reflected when comparing the performance of their second model to the first. On average, participants achieved a 0.746 ($SD = 0.38$) F-score the first time and a 0.749 ($SD = 0.28$) the second with no significant change ($W = 80.5, Z = -0.16, p = .871$). However, participants who indicated they would do nothing to improve their model after the first attempt ($N = 22$), seem to achieve significantly higher performance than the rest ($U = 1472, Z = 5.22, p < .001, r = 0.52$) and this is a consistent trend across both attempts ($U = 1459.5, Z = 5.12, p < .001, r = 0.51$). Looking at these relative low F-scores for such a simple 3-way classification task, it is surprisingly that the second group of participants did not further improve their performance even though they expressed reasonable strategies. Perhaps the incentives were not strong enough and they had a higher threshold for errors, or there was not enough time and iterations to try things out. It could simply be that their object instances were too similar. Indeed, the majority (N=38) of the participants in this group had chosen spices.

8.4 Discussion

We see how our results, some being new insights, others strengthening prior empirical and anecdotal evidence, can help better understand non-experts' interactions with machine teaching and guide the design of future teachable interfaces. We highlight some of them with the following suggestions:

Account for teaching strategies: Our observations suggest that non-experts mainly tend to teach with clear representative examples and sometimes incorporate examples that are closer to the decision boundary through variation, which draws from parallels to

how humans generalize for similar recognition tasks. In the case of object recognition, these were object size, viewpoint, location, and illumination [158]; though all four were considered only by a few. Our analysis also suggest that beyond class imbalance [155, 157], there can be other disparate treatments such as inconsistency in the way variation is incorporated across classes.

Anticipate misconceptions: A prevalent misconception relates to consistency. While it is true that consistency between training and testing data will result in better performance, assuming they both represent real-life examples, some thought that being consistent entails teaching with multiple identical examples with no variation whatsoever. Other misconceptions relate to the capabilities of the machine for reasoning. For example, participants would train with visually disparate examples from both the container and its content separately. Others would assume that the models were able to infer the text.

Help users craft evaluation examples: Our observations indicate that testing examples tend to be less diverse or not at all. Thus, it is no surprise to see many people wanting to change nothing, being satisfied with the performance, or not knowing what to do. Even those who did change their behavior when training for a second time, it was to not vary the background rather than making their model more generalizable. Help may look different based on the goal of the teachable interface. If it is personalization (*e.g.*[195]), then it could mean guiding the user to generate examples that are more representative of future use cases [155]. However, if it is an application intended to uncover machine learning concepts (*e.g.*[151]) perhaps promoting more model-breaking examples [196] would be more appropriate; though in the context of a teachable interface this could lead to users training the model with less authentic data to simply improve its performance [157].

This work has several limitations listed below:

Task: We explore machine teaching in a narrow context, that of a supervised 3-way image classification task. This allows us to dive deep in our analysis using a fine-grained scheme when coding participants examples informed from prior work on visual object understanding. However, it also limits the generalizability of our findings. We attempt to overcome this by connecting our results with that of prior work when possible. Three, the smallest number for multiclass classification, was selected to minimize challenges in finding different object instances within a category in a real-world environment as well as the task completion time (already 40 minutes long).

Study: While teachable object recognizers are real-world applications [146], they are typically intended for blind users. Thus, the sighted participants may lack motivation in this study. We attempt to compensate for this lack of incentives with a performance-based payment scheme [182] creating the impression that we have a 'secret' test to distinguish models that are more 'robust'; though on our end this is merely a naive quality examination. By doing so, combined with the fact that the testbed shows only the predicted labels but no confidence scores in testing, we might have limited participants' criteria for model evaluation [155] to just correctess.

Analysis: Through crowdsourcing we were able to quickly recruit a large participant pool and collect data outside a lab in the users' environment. However, this limited our control over the object instances that participants could use as well as the opportunity to create our own evaluation set for comparing the performance of the models against the same data.

To allow some time before testing for the photos to be received on our server and the

models to be trained on our GPUs, participants were asked to review their training photos and select 10 out of 30, 5 out of 10, and 1 out of 5. We are still analyzing these data while considering more fine-grained variation and inconsistency attributes.

8.5 Conclusion

We have presented a crowdsourcing study, where MTurkers choose three objects in their environment and iteratively train a model to distinguish between them in real-time using the camera on their mobile phones. By doing so, we were able to explore, with a large participant pool ($N = 100$), an instance of a machine teaching problem with a task where many non-experts can serve as the oracle. Our findings and insights can contribute to the ongoing discussion on how non-experts conceptualize, experience, and reflect on their engagement with machine teaching. To allow for study replicability and future comparisons, we have provided a detailed description of our testbed, its framing within the machine teaching problem space from Zhu*et al.* [4], and the list of questions and prompts used in the study.

Our results are based on a fine-grained analysis of the participants' examples contextualized by their responses, background, and model performance. We discuss how they can guide the design of future teachable interfaces to anticipate users tendencies, misconceptions, and assumptions. Given our research group's interest in teachable interfaces for accessibility [195], our next step will be to explore whether these insights and data from sighted participants could be leveraged for the design of effective teachable object recognizers for blind users. Our rationale is that insights from this study can perhaps enable us to decouple non-

experts misconceptions from challenges in camera manipulations among blind users [146].

Chapter 9: Designing a Teachable Object Recognizer with Training Set Descriptors for Blind Users

9.1 Motivation and Introduction

In Chapter 8, we defined some attributes of photos that would affect the performance of a teachable object recognizer. We see that the attributes describing the photos and teaching strategies among sighted users can be leveraged to serve as descriptors in teachable object recognizers where descriptors inform blind users of the attributes of their training photos. To demonstrate this implication, we built TOR, an accessible teachable object recognizer that enables blind users to review their training photos through a set of descriptors.

In this chapter, we design, implement the TOR app with descriptors, and evaluate it with blind participants through a simulated controlled study that was conducted in blind participants' homes due to COVID-19. The user study explores the blind participants' experience with the TOR app and descriptors by asking blind users to use a prototype of the TOR app and asking questions about their experiences. In addition, while the user study in Chapter 8 explores the patterns of training and testing a personal object recognizer with sighted participants, the user study in this chapter further examines the interactions of blind users and a mobile personal object recognizer app more thoroughly

including the tasks of training, testing the app, managing items, and iterating these processes to improve the app. We analyze the blind users' training and testing strategies based on their photos taken during the tasks and subjective feedback after the tasks.

We report on a user study with 12 blind participants. We found that the descriptors in the TOR app benefited the users in their experimentation for collecting good training examples. Though the descriptors provided approximately estimated attributes of photos with some errors, the participants could understand the attributes that would affect the performance of TOR and inspect their training examples with the descriptors. With the interface design based on findings from prior studies, the subjective evaluation by the blind participants showed that they could effectively train the object recognizer, test it, and manage the information of the objects in their training sets. However, they pointed out important design issues that should be resolved in a future study such as the time-consuming process to collect many photos for training. The accuracy of recognizing objects with TOR trained by the blind participants was only 0.65 which is low considering that we included only three objects in the study, though, they were engineered for a worse-case scenario. We identified the possible reasons for this through the analysis of the participants' photos, revealing that they had a lack of variation, cropped objects, cluttered backgrounds in their training sets as well as test examples.

To the best of our knowledge, this is the first work to propose non-visual access and to provide empirical results with blind participants on automatically estimating and incorporating accessible descriptors for inspecting training data in teachable computer vision applications. Our analysis focuses on object recognizers, where 'learning to train' is deemed as one of the main challenges among blind users [20, 195]. However, we see

how the underlying methods for extracting meaningful instance- and set-level descriptors can be adopted for other teachable assistive technologies. Perhaps, they can also serve as the first step towards more accessible approaches for explainable AI interfaces, where there is an underlying assumption on people's ability to visually inspect explanations [197, 198, 199].

9.2 Method

We built a prototype of TOR app on Apple iPhone 8 and evaluated the app design through a user study with blind participants.

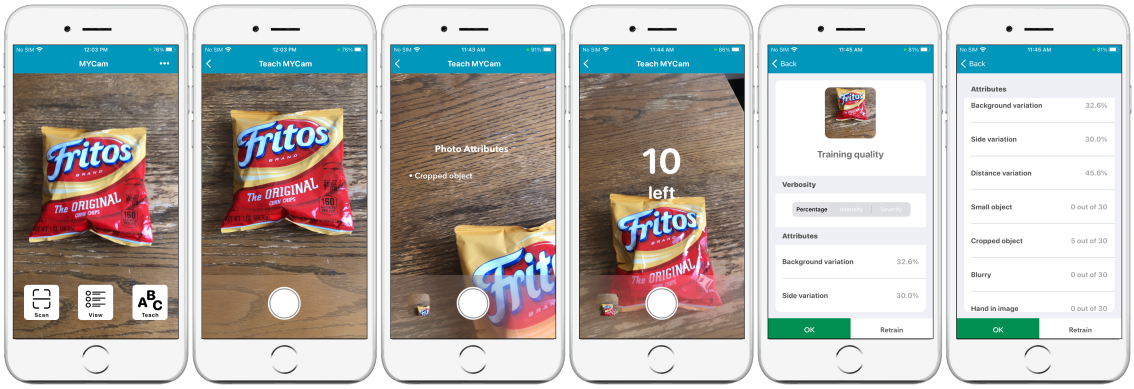

Figure 9.1: Screenshots from the TOR app indicating from left to right the home screen, teach screen, teach screen with descriptors, teach screen with the number of remaining photos notification, review screen (top), review screen (bottom).

9.2.1 Interface Design

As shown in Figure 9.1, when users open the app, they enter the main screen which has three buttons, *Scan item*, *View items*, and *Teach TOR* (Figure 9.1). Users can recognize an object, manage items in the training dataset, and collect photos of an object of interest for training the recognition model, respectively.

9.2.1.1 Training

The Teach TOR button brings users to a screen where they can take photos for training an object recognition model. The training process includes three steps: taking photos, reviewing the training examples with descriptors, and providing information (*i.e.*, labeling and recording an audio description) for the object.

Taking photos. When users select the Teach TOR button, the app displays a camera screen where users can take a photo using the shutter button at the bottom center (Figure 9.1). As soon as they take a photo, the photo is sent to a back-end server which calculates the descriptors of the photo that may affect the performance of an object recognition model. The descriptors are instance- and set-level attributes (Table 9.1) devised to help blind users examine their photos in terms of their quality for the purpose of training. As soon as a photo is taken, users are notified of the instance-level attributes (*e.g.*, a cropped object in the photo) with synthesized speech. The attributes are visually displayed on the screen at the same time as shown in Figure 9.1. For every object, users take 30 photos with the count indicated in real-time.

Reviewing the photos with descriptors. As shown in Figure 9.2, when users are done taking the 30 training photos, the app presents a screen with the set-level attributes indicating how much variation there is among the photos in terms of object size and perspective as well as background. More so, an aggregate of the instance-level attributes is given indicating the total number of photos where the object is too small or cropped, the image is blurred, and the hand is present. After reviewing the descriptors, users may select *OK* to proceed or *Retrain* to restart the training process.

Instance-level attributes	
Small object	The bounding box of the object is smaller than 1/8 (12.5%) of the image.
Cropped object	The object is partially included in the image.
Blurry photo	The photo is too blurry to recognize textures or texts.
Hand in photo	A user's hand is visible in the image.
Set-level attributes	
Variation in size	A set of images shows objects with different sizes.
Variation in perspective	A set of images shows different sides of objects.
Variation in background	A set of images show backgrounds with different textures or items.

Table 9.1: Our descriptors for reviewing photos are informed by prior studies exploring how people who have no machine learning expertise synthesize their data for training and iterate on them when they can access them visually [1, 2**?**].

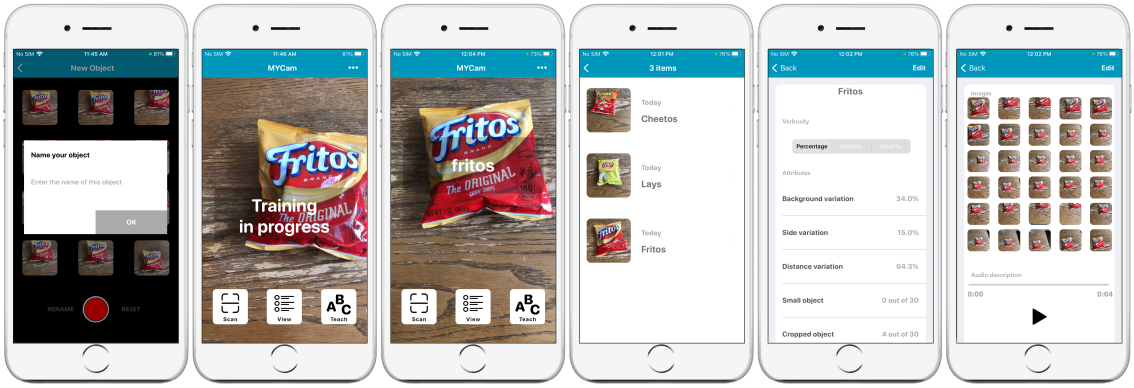

Figure 9.2: Screenshots from the TOR app indicating from left to right the labeling screen, home screen when training is in progress, home screen with a recognition result, list of items screen, item information screen (top), item information screen (bottom).

Providing information about the object. Before the object recognition model is trained with the photos, users need to provide a name and optionally an audio description for the object. A dialogue box with a text field shows up so that users can enter the name of the object which will be used as a label for training (Figure 9.2). Once this step is completed, the app notifies that training has started. At this moment, the object recognition model is trained with the photos on the server-side. While training is in progress, the Scan item and Teach TOR buttons on the main screen are disabled. Users

are notified when training is done.

9.2.1.2 Recognizing Objects

The main screen shows a camera view to allow users to take a photo with the Scan item button. The Scan item button is disabled when users have trained the object recognition model with fewer than three objects. After training the model with three or more objects, users can recognize objects by taking photos of them with the Scan item button. When a photo is taken, the photo is sent to a server where users' personal object recognition models make a prediction. The mobile app plays a synthesized speech of the label and visually displays it on the screen. Users hear the label in 100 milliseconds after taking a photo. To distinguish the objects not in the training dataset, we employed an approach of quantifying the confidence level of the discriminability with the entropy of confidence scores [200]. Specifically, when the entropy value is greater than 2.0 or the confidence score is lower than 0.4, the app says "Don't know" in synthesized speech instead of the label from the model. The thresholds of the entropy and confidence score were decided based on internal tests conducted by our research team.

9.2.1.3 Managing Items in One's Dataset

When users select the View items button on the main screen, the app shows a screen with a list of items (Figure 9.2). The list includes the names of objects, dates when the objects are added, and thumbnail images. When users select one of the items, the app brings them to a screen with descriptors and the photos that the users have taken to train

the object recognizer (Figure 9.2). In this screen, users can select the edit button to change the name of the object and re-record the audio description as they did when entering the name and recording an audio description of the object for training (Figure 9.2).

9.2.2 Implementation

We build TOR as an iOS app on the Apple iPhone 8. For the object recognition and estimation of the descriptors, we use various computer vision techniques such as image classification, object detection, and hand segmentation. To speed up the calculations for real-time interactions, these functions run on a back-end server with GPUs, though, the promise of teachable object recognizers is that eventually they will run on the device for more privacy. The TOR app and the server communicate through HTTP.

9.2.2.1 Object Recognition Model.

The base model for object recognition is Inception V3 pre-trained on ImageNet [5]. When users train the TOR app, the last layer of the base model is fine-tuned using transfer learning with the photos taken by the users. The transfer learning is done with a gradient descent algorithm with 500 iterations and a 0.01 learning rate. For example, the training takes around 80 seconds with 90 photos of three objects.

9.2.2.2 Descriptors.

The attributes in Hong *et al.* [201], which inspired our descriptors, were originally coded manually by two researchers after visually inspecting the photos taken by the

participants. Given that this is not a trivial process, methods like Wizard of Oz did not deem appropriate in this early exploration of these descriptors for facilitating an experimentation that is accessible for blind users. Thus, we opted for methods that attempt to automatically estimate them, even though, developing techniques for more accurate estimations is beyond the focus of this paper and is briefly discussed in section **??**. In the current version of TOR, the descriptors are estimated with the following approach:

- **Small object**: The bounding box of an object in the image is detected by a YOLOv3 object detection model [202]. The object is considered too small if the size of the bounding box is smaller than 1/8 (12.5%) of the image.

- **Cropped object**: If the bounding box is at the edge of the image, we considered the object cropped.

- **Blurry photo**: An image is converted to a grayscale image (the values of pixels have a range of 0-255). If the variance of pixels in the output of Laplacian edge detection [**?**] is lower than 3.0, the photo is considered blurry.

- **Hand in photo**: The pixels from a hand are detected via a hand segmentation model that has been previously tested with blind participants [140]. If the pixels are more than 0.3% of the image, a hand is detected.

- **Variation in size**: The position of the camera (*i.e.*, the smartphone device) is detected using the 3D coordinate system in ARKit[1] of iOS when a photo is taken. As the size of the object changes depending on the distance between the camera

[1] `https://developer.apple.com/documentation/arkit/content_anchors/scanning_and_detecting_3d_objects`

and the object, the differences in the positions of the camera are measured. The variation is calculated using the standard deviation of the differences.

- **Variation in perspective**: The sides of an object are detected using the 3D object detection in ARKit. The number of sides shown in the photos is used to measure the variation in perspective. While the object can be detected with ARKit when the object is captured clearly with a plain background, it would not work with photos with cluttered background or a cropped object. When the object is not detected from a user's photos, the orientation of the camera was used to measure the variation in perspective instead of the number of sides assuming that a user would move the camera to capture different sides of the object. We calculated the standard deviation of the cosine similarities between the orientations in the 3D coordinate system of ARKit to measure the variation in perspective in this case.

- **Variation in background**: Assuming that the backgrounds captured in photos can vary as a user moves the camera to different places or change its orientation, we used the orientation and the location of the camera to measure the variation in the background. Both the standard deviation of differences in orientations and locations in the 3D coordinate system are calculated. The maximum value of the two standard deviations is selected as a variation in background.

9.2.3 Procedure

To explore the usability of TOR and its potential for increasing the accessibility of experimentation with teachable object recognizers, we conducted a user study with

blind participants. Participants were asked to train the app to recognize three snacks. Participants carried out tasks of training, testing their object recognition models, and reviewing the information of the items in their training dataset with the app. After each task, we asked questions about their experiences. The study is approved by IRB at [Anonymized institution] (IRB #: anonymized).

9.2.4 Participants

We recruited 12 blind participants (6 female, 6 male) from campus email lists and local organizations (Table 9.2). The participants ranged in age from 32 to 70 ($M = 54.3, SD = 15.2$). They self-reported being totally blind ($N = 3$), having some light perception ($N = 5$), or being legally blind ($N = 4$). All participants have used smartphones several times a day. P1 and P2 reported having some hearing loss (auditory processing disorder and difficulty in hearing high frequencies, respectively). All participants reported that they take a photo or record a video at least once a month. When asked to report their levels of familiarity with machine learning in 4-scales: not familiar at all (have never heard of machine learning); slightly familiar (have heard of it but don't know what it does); somewhat familiar (have a broad understanding of what it is and what it does); extremely familiar (have extensive knowledge on machine learning), two participants selected not familiar at all, eight selected slightly familiar, and two reported being somewhat familiar.

ID	Age	Gender	Level of vision	Age of onset	Familiarity with ML*
P1	39	Female	Light perception	Birth	Not familiar at all
P2	67	Male	Legally blind	55	Slightly familiar
P3	62	Female	Totally blind	Birth	Somewhat familiar
P4	32	Male	Legally blind	20	Slightly familiar
P5	66	Male	Light perception	46	Slightly familiar
P6	61	Male	Light perception	41	Somewhat familiar
P7	70	Male	Legally blind	Birth	Slightly familiar
P8	50	Female	Legally blind	45	Slightly familiar
P9	69	Female	Totally blind	55	Not familiar at all
P10	66	Female	Light perception	Birth	Slightly familiar
P11	33	Female	Light perception	Birth	Slightly familiar
P12	36	Male	Totally blind	Birth	Slightly familiar

*ML: Machine learning

Table 9.2: Participants' characteristics.

9.2.5 Procedure

The study took place in participants' homes due to COVID-19. The participants were given a laptop and wore Vuzix Blade smart glasses [**?**] with an online meeting application (Zoom) to communicate with the experimenter remotely. The study consists of three tasks: 1) training the TOR app with photos of objects, 2) testing the performance of the TOR app, 3) reviewing and editing the information of the objects. At the beginning of the study, we explained the concept of TOR briefly with minimal description of how to take photos to train or test the app effectively so that we can observe participants' strategies for taking photos for training and testing an object recognizer. The description of the app given at the beginning of the study reads as follows:

> "The idea behind the app is that you can teach it to recognize objects by giving it a few photos of them, their names, and if you wish, audio descriptions. Once you've trained the app and it has them in its memory, you can point it

Figure 9.3: Object stimuli in the study: Fritos, Cheetos, and Lays.

to an object, take a photo, and it will tell you what it is. You can always go back and manage its memory."

Participants were asked to train the app with photos of three objects in Figure 9.3. The order of objects was fully counterbalanced between participants. When participants train the app with the first object, the experimenter provided instructions on the user interface of the app step by step. For the second and third objects, participants were asked to train the app for themselves and they could ask the experimenter about the user interface if necessary. After training the app with three objects, they tested the performance of their models by taking photos of the objects. Participants did not have any restrictions on how many photos they need and how the objects should be captured during the tests. They measured the performance of their object recognizer and decided when to finish testing it for themselves. After the tests, participants were asked to review the information of an object (*i.e.*, descriptors, label, and audio description) and edit the label of it at the end.

Throughout the study, we encouraged participants to think out loud and to ask questions at any time. For each task, we asked questions related to the experience with teachable interfaces and usability satisfaction questions developed by Lewis [**?**]. At the end of the study, we also had a post-task interview with open-ended questions about their overall experience with the app. All questions in this study were either open questions or

on a 5-point Likert scale (*i.e.*, strongly disagree, disagree, neutral, agree, strongly agree).

9.2.6 Object Stimuli

Based on the need for recognizing objects with similar sizes, weights, and textures [20] with fine-grained labels, we used three snacks for this study with the same size, texture, and nearly identical weights, shown in Figure 9.3. With these snacks, we could simulate a scenario that a blind user uses TOR to recognize different objects that are difficult to distinguish with the tactile sensation only. It is engineered to be a challenging scenario as it involves fine-grained recognition for similarly shaped and colored deformable objects with reflective surfaces. Unique and personal objects without logos or texts on them (*e.g.*, key, mug cup) can be potentially used with TOR and perhaps could fit a more realistic scenario. However, for this study, we included only commercial products to allow for comparison and replicability similar to what other prior studies regarding object recognition have done [20, 146, 203].

9.3 Results

We found that participants could train an object recognition model, test to see how well it works, and review the information of their training examples with TOR. All participants could complete the tasks in the study successfully. While participants' responses to the questions in the study revealed that it was easy to complete these tasks in general, they also pointed out some design issues to improve the usability of the app. Moreover, analysis of participants' feedback and photos revealed some problems in their

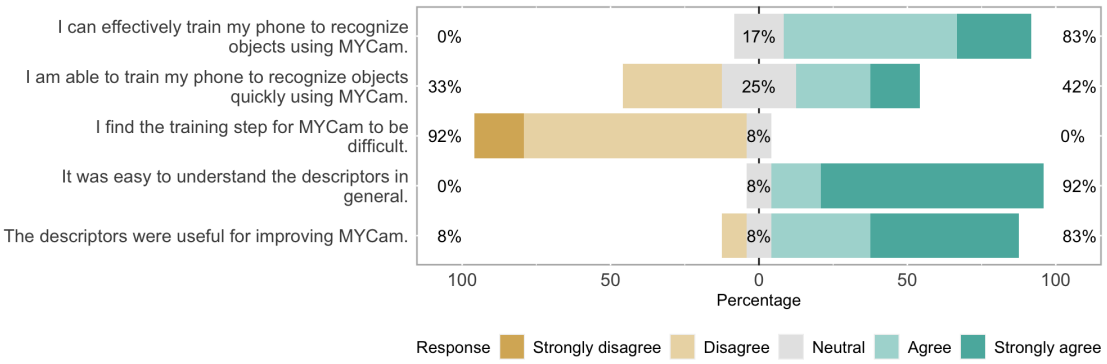

Figure 9.4: Participant responses to questions about their training experience during the study.

strategies of taking photos for training and testing an object recognition model which can be potentially resolved with guidance from the app.

9.3.1 Training

We evaluate the interface design based on participants' feedback. We analyze the interaction with descriptors and participants' strategies of training an object recognition model in-depth.

9.3.1.1 Interacting with the Interface for Training

Participants spent 143.8 seconds ($SD = 72.4$) to take 30 photos of an object on average. Six participants re-trained the app with at least one object after reviewing the training photos with descriptors. All participants completed the training task though the performance of the object recognition model varied across participants. When asked if they could train the app effectively, ten participants agreed that they could. Seven participants thought the training interface was easy to learn and straightforward. P1 and

P10, for example, who are not familiar at all and slightly familiar with machine learning said *"after a while, I learned that I could train it"* and *"It's pretty easy. You have to teach me though. But if you teach me then it's pretty easy to follow instruction and finish the process,"* respectively. Three participants mentioned that the descriptors helped them understand their training examples. For example, P3 said, *"(I could train it effectively) because you can use the feedback for determining if you've gotten a good representation of the object."* On the other hand, P11 and P12 neither agreed nor disagreed that they could train the app effectively. P11 pointed out that taking 30 photos is a time-consuming task, saying *"I don't really feel like I was all that effective, because it takes a while to train for each one."* P12 thought the descriptors were not helpful due to errors. P12 mentioned *"I don't think that the app is correct, especially when I know, for example, that my hand was not in the photo...I don't have a lot of confidence in the app's accuracy."*

When asked about whether they could train the app quickly, five participants agreed, but four disagreed and three were neutral. Seven participants thought that taking 30 photos is tedious. For example, P10 said, *"The process is pretty straightforward. But I have to spend, like, quite long time to train the three objects."*. To resolve this problem, P6 suggested allowing users to record a video to shorten the step for collecting multiple photos of an object. For the question about the difficulty of the training task, all but one participant agreed or strongly agreed that the task was not difficult. P11 who was neutral thought it was not difficult but tedious.

While participants could record audio descriptions if they want, only four participants did during the study. One of the participants added the audio description to clarify the object with details. P10 said *"The reason I did with the Lays is because Lays makes*

different things. They make potato chips, Fritos. So I wanted to say it was potato chips because I thought that's what they were." Six participants thought that the label itself was enough to understand the object. However, three of them mentioned that they would add audio descriptions for other kinds of objects. P7 said, *"[...] regarding some food, if it were, for example, milk, I might add it for any of the other items that have an expiration date."* Another reason for not adding the audio description was that two participants did not like listening to their own voice from the app. On the other hand, P2 thought it is easier to understand his own voice than synthesized speech.

9.3.1.2 Interacting with the Descriptors

All but one participant (P1) agreed or strongly agreed that the descriptors are easy to understand. This indicates that the factors that may affect the performance of the object recognizer are easily understandable to the users. P6 said *"I understood what it was telling me. I didn't have questions about what I was supposed to do."* However, the values of descriptors presented while the participants reviewed the photos would be somewhat ambiguous to the participants as mentioned by P1 who was neutral on this. P1 said, *"I guess just knowing exactly what they're referring to what numbers are really preferable."* P4 also mentioned the challenge in understanding the values, but he could figure it out based on his experience during the study. P4 said *"I wasn't aware of any of those fields when we did the first object [...] For the second and third objects. I could take a little bit more variation in the photos or to better train the application."*

While the current app had some errors in estimating the descriptors of photos,

participants thought the descriptors are useful to understand what to do to collect good training photos. To measure the accuracy of the descriptors, we calculated the correlation coefficients between the estimated and manually annotated attributes. The estimated attributes are the percentages of variations and the number of photos with instance-level attributes estimated by the app during the study. A researcher manually annotated the photos from the participants based on the definitions of the descriptors in the section 9.2.2.2. To quantify the variation of background and perspective, the researcher grouped the photos in a training set with the same background and the side of the object. The groups are used to calculate the Shannon-Wiener Diversity Index [204]. The cropped object, hand-in-photo, and blurry-photo attributes are simply quantified as the number of photos with the attributes identified through visual inspection. For the attributes related to the size of the object (*i.e.*, variation in size, too small object), the researcher annotated the bounding boxes of the objects. The manually annotated variation of size and too small object attributes are the standard deviation of the sizes of the bounding boxes which range from 0.0 (*i.e.*, the object is not captured) to 1.0 (*i.e.*, the size of photo) and the number of photos with bounding boxes smaller than 12.5% of the photos.

The low correlation between the manually annotated and estimated attributes (Figure 9.6) shows that the descriptors had some flaws. The correlation coefficients between them ranged from 0.23 to 0.57. There were no photos estimated as having too small objects by the app while the manually annotated bounding boxes in three photos in Figure 9.5 were smaller than 12.5% of the photos due to the cropped or obscured objects. While we employed naive approaches for estimating the descriptors as a proof of concept, generating accurate descriptors is a complicated problem. Some possible reasons for the

Figure 9.5: Training photos annotated as having too small objects (target objects are marked with blue dotted rectangles).

low correlations are the poor object detection in idiosyncratic environments (*i.e.*, textures of backgrounds, light conditions, cluttered photos) and mismatch between the movement of a camera and the visual changes in photos. We discuss the technical problem in estimating attributes in the section **??**. While descriptors had some errors, ten participants agreed or strongly agreed that the descriptors were useful. P10 and P11 thought descriptors helped them understand how to collect training examples for the object recognizer. P10 said, *"(I agree) because I know the quality of the photos, the different aspects of the photos that I take."* P11 said, *"It helped me understand what the camera needed in order to recognize the objects."* Participants also used them to diagnose problems in their training sets. P10 elaborated *"you have to get feedback or you're not going to improve [...] it helps you to understand what you're doing wrong."* P2 had a similar idea: *"the explanation afterward, in the analysis, told me that my photographs were not always good, so I have to learn to take better photographs."* On the other hand, P11 neither agreed nor disagreed that descriptors are useful. P12 thought they were not useful because of the errors. P12 said, *"I don't think that the app is correct, especially when I know, for example, that my hand was not in the photo, or that the object is not cropped because the previous objects were cropped."*

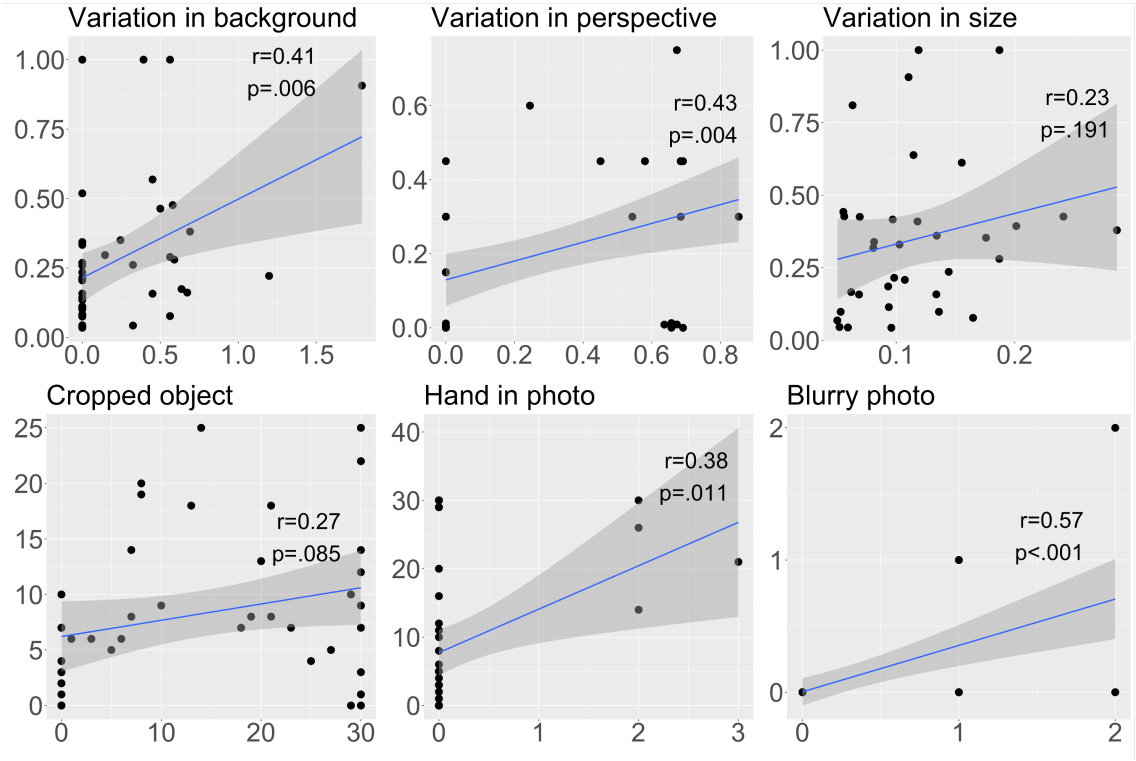

Figure 9.6: Scatter plots with the manually annotated values on the x axis and estimated values on the y axis. The correlation coefficient (r) and p-value (p) are specified in the plots.

Participants thought that specifying how to resolve issues the descriptors. P7 suggested integrating the feedback for blind photography (*e.g.*, [141, 146]) and descriptors, elaborating *"Cropped, it did not help me know what to do differently. If it said, maybe move up, move down and move camera left, move the camera, right. That would have been more useful."* P6 mentioned that the interface for replacing problematic photos in a training set would improve the app. He said *"I would assume the training process can self-evaluate itself and it should sum that up for me and tell me what photos I should replace. [...] you need to replace those bad pictures unless you don't need them for the training."*

9.3.1.3 Training Strategies

When participants finished the training task, we asked a question about what they thought was important to consider when training the object recognizer. The most frequent responses were about diversifying the photos. Five participants thought varying the distance between the camera and the object was important. Another five participants intended to vary the perspectives in photos. Some reasons for variations were: they wanted to include photos that would be similar to the photos they would take for recognition with different perspectives and sides; they hoped the app to learn the visual information from different sides of the objects. Four participants mentioned that centering the object was important. The participants' responses revealed that the descriptors affected their strategies in collecting photos. For example, P3 said *"It was important to consider the instances of cropped photo and handed photo. You know, it was always good to hear when it would just click the shutter and then not hear those two things."* and P7 mentioned, *"you can make adjustments very easily so there's a good chance you're going to get a reasonable percentage that would help you to identify the object."*

We also analyzed the photos from the participants based on the manually annotated descriptors to identify patterns and problems in the photos. The majority of participants varied photos in their training sets with at least one object. Eight and six participants varied the background and perspective, respectively (*i.e.*, the diversity index is greater than 0). Eight participants varied the size of objects captured in photos (*i.e.*, the standard deviation of the sizes of bounding boxes is greater than 0.1). On the other hand, we found that seven participants took photos with no variation at all with at least one object

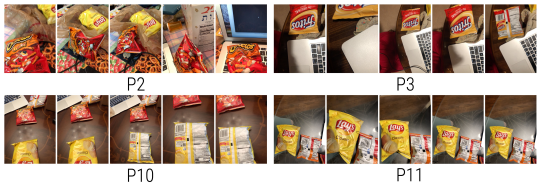

Figure 9.7: Training photos with cluttered backgrounds.

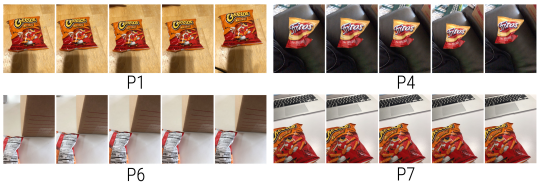

Figure 9.8: Training photos with little variation.

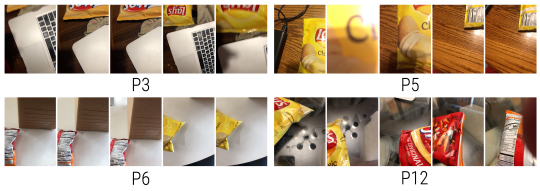

Figure 9.9: Training photos with problems in framing (*i.e.*, adjusting the distance and centering the object).

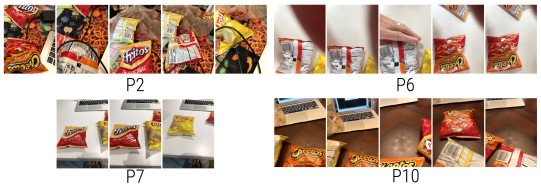

Figure 9.10: Test photos with cluttered backgrounds.

(*i.e.*, the diversity indices of the distance and perspective variation are 0, the standard deviation of sizes of bounding boxes is lower than 0.1). The example photos with no variation are shown in Figure 9.8. This is consistent with the findings from a prior user study on exploring non-experts' perception of machine teaching [**?**]. It showed the majority of non-experts in machine learning are aware of the importance of diversity in a dataset though having no variation at all is frequently observed in their photos. We also observed quality issues in participants' training samples. Four participants took photos with cluttered backgrounds as shown in Figure 9.7. The training examples from ten participants included photos with poor image framing (*i.e.*, Cropped object) as shown in Figure 9.9

9.3.2 Testing (Recognizing Objects)

We evaluated the interface design for testing (*i.e.*, recognizing objects) based on the responses from participants. The responses and photos revealed patterns in how users test an object recognizer and interpret their test results.

9.3.2.1 Interacting with the Interface for Testing

After testing the app, we asked participants if they could test the object recognizer effectively and quickly. These questions are about the effectiveness of the interface and time for understanding the performance of the object recognizer regardless of its performance. Ten participants agreed or strongly agreed that they could test the object recognizer effectively and quickly. Most participants thought it was just easy and straightforward. P10 said *"I felt like I went through it pretty quick. I felt like I understood what to do."* On the other hand, two participants, P2 and P3, who disagreed pointed out that the misrecognitions in the tests made it hard for them to evaluate the app. P3 said *"(I disagree because) I got different results (with one object). I'd want to be certain about what I was getting."*

9.3.2.2 Strategies for Testing the App

As prior studies showed that it is challenging for non-experts to test a machine learning model systematically [20, 157, 205], the analysis of participants' testing samples revealed some patterns that may be problematic in conducting a thorough evaluation of an object recognizer. During the testing task, participants took 3.7 photos per object

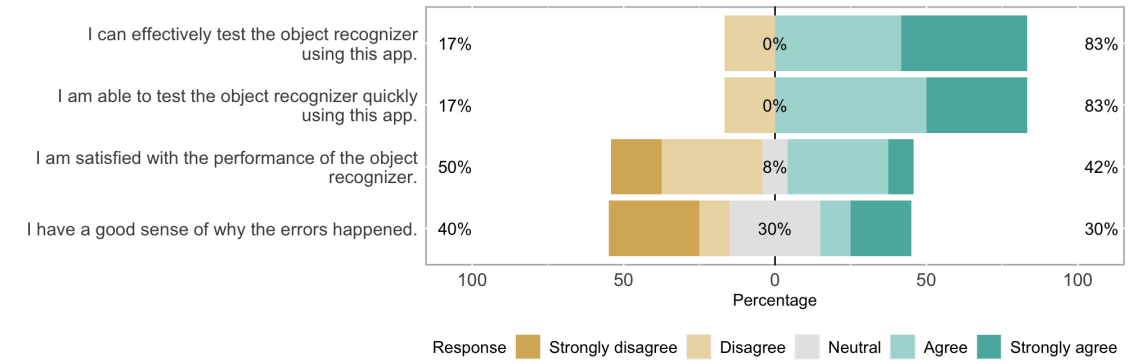

Figure 9.11: Participant responses to questions about their testing experience during the study.

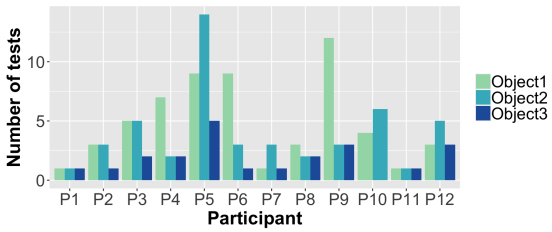

Figure 9.12: The number of tests per object.

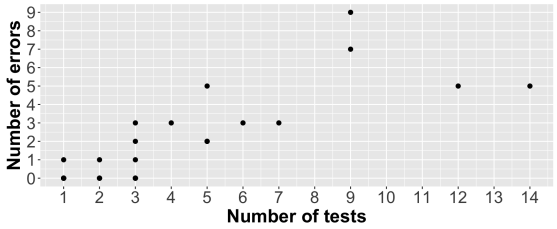

Figure 9.13: The proportion of errors and number of tests.

Figure 9.14: The number of tests per object and proportion of errors.

on average ($SD = 3.2$). Considering that the test data need samples with different visual contexts (*e.g.*, sides, sizes of objects, background, light condition) to test the object recognizer thoroughly, participants would have had fewer photos for testing than necessary for a thorough evaluation. Looking at the number of test photos per object (Figure 9.12), we observed that the number of test samples was different across objects. The number of errors would affect the number of tests as the correlation between the number of errors and test samples per object (Figure 9.13) is strong (Pearson correlation, $\rho = 0.82, p < .001$).

The average accuracy (*i.e.*, the number of correct predictions divided by the number of total test samples) of the object recognition models was 0.65 ($SD = 0.24$) when they

are tested by the participants (Figure 9.15). While machine learning models are typically evaluated with large benchmark datasets, the test sets from participants include photos from participants' idiosyncratic environments. Looking into the test photos, we found that the test photos had quality problems that would affect the validity of the participants' evaluations. One of the frequent problems were about image framing. The test photos from Four participants included less than half of the objects. We also observed that four participants took photos capturing two or three snacks, making it hard for the app to distinguish which one they wanted to recognize 9.10. The problems in the test sets would be critical as the perceived and actual performance of TOR may be different when it is used after training.

9.3.2.3 Interpreting the Test Results

When we asked participants if they were satisfied with the performance of the object recognizer, five participants agreed or strongly agreed, six participants disagreed or strongly disagreed, and a participant was neutral. Looking at their responses and the performance of the object recognizer together (Figure 9.16, we observed that participants were not satisfied if the accuracy was lower than 0.6. On the other hand, the accuracy spread between 0.6 and 1.0 with participants who were satisfied with the performance. Based on the responses from the participants, we found that performance was not the only factor that affects the users' satisfaction. One of the factors was the amount of effort for training the object recognizer. P11 was neutral though she did not observe any errors because the training task was tedious. P11 said, *"Because it took so much work to get*

that small amount of performance." P7 and P10 agreed that they were satisfied with the performance though the accuracy was only 0.6 and 0.4, respectively. P7 has low vision and it is enough to supplement his vision. P10 thought she did not train the app properly with an object that the app made misrecognitions with. P10 said, *"I think it recognized objects, but if you don't train it properly, then it's not going to recognize anything [...] the Fritos bag was the one that didn't work out, but that was probably my fault."*

While nine out of the 12 participants observed misrecognitions during the tests, the participants mostly did not have any idea of why they happened during the tests. Six participants were neutral or disagreed that they have a good sense of why the misrecognitions happened. Their responses were simply *"I have no idea."* or *"I don't know."* Though P7 and P10 strongly agreed and agreed, respectively, they had abstract ideas about the errors. P10 said *"I think it was my fault. I think it was my training. Other than that, I don't know."* P9 strongly agreed because the descriptors provided feedback that the samples had problems, elaborating *"The reason is because I was teaching it, and I wasn't 100% sure that it was 100% accurate. It makes sense that while I was teaching it, I was a little bit off, so its recognition was a little bit off. It kept telling me that the hand was in the photos."*

9.3.3 Managing Items in the User's Dataset

All participants successfully reviewed the information of objects (*i.e.*, listening to descriptors, audio descriptions, labels) and edited the label of an object. This would be because the task consists of basic interactions used in other apps such as navigating

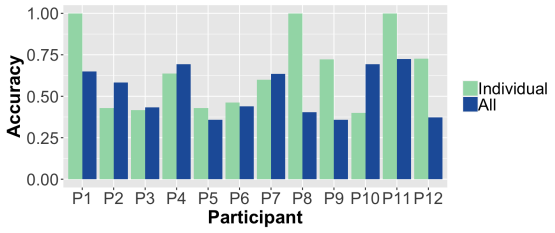

Figure 9.15: The accuracy of the object recognition models tested by the participants.

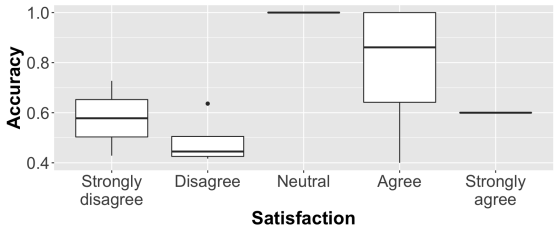

Figure 9.16: Average accuracy versus satisfaction with the performance. The red dots are means.

Figure 9.17: The number of tests per object and proportion of errors.

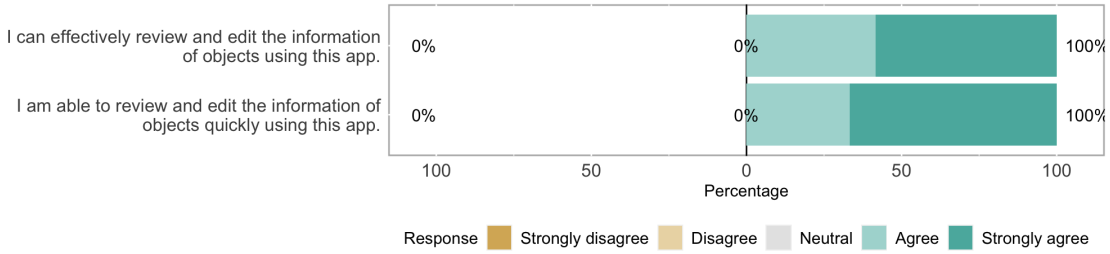

Figure 9.18: Participant responses to questions about their reviewing and editing experience during the study.

through a table view and entering texts in a text field. They also agreed or strongly agreed that they could complete the task with this app effectively and quickly. Participants said *"It was easy to do"*, *"That was easy"*, *"the steps to be taken on the app was easy to follow."* P12 thought the interface can be improved by integrating the review and edit interface. He elaborated *"Because it's pretty easy, although it could be more streamlined. I would think that you could put the name right on the review screen just as a text field."* P6 pointed out that the difficulty of using this interface would depend on the experience with the iPhone as the interface of the app use typical designs and controls in iOS apps. P6 said, *"I guess if you never used an iPhone before, it might be a little bit of an effort but the interface is compatible with voice ever."*

9.3.4 Overall Experience

At the end of the study, all participants agreed or strongly agreed that it was easy to learn to use the TOR app. Nine participants agreed or strongly agreed that it was simple to use the TOR app overall while two participants were neutral and one disagreed as shown in Figure 9.19. The three participants thought the training process is inefficient. They pointed out that taking 30 photos for training is tedious and inefficient. In particular, P12 who disagreed could not come up with any case that training would be necessary assuming that users know about an object when they train the app. P12 said *"The very fact that I have to teach it makes it inefficient. If I have to teach it when an object is, then I already have to know what an object is."* Based on the responses from participants, we found that they have different attitudes toward a teachable interface. For example, P11 thought the effort for training is a lot, considering that the information from the object recognizer is small. P11 said *"I honestly feel like it takes too long to do that. I feel like if you have to train it to recognize things, you're not going to be as efficient. I like the other way better, where you just have it read the label (using a text recognizer)."* On the other hand, P9 was positive about having a teachable interface for an object recognizer. She said *"To identify what an object is so handy. And then to be able to teach it you know, to identify items that may not already be there is particularly powerful because you know, something's not there, you have the ability to include it."*

All participants thought that the organization of the interface is clear. All but one participant agreed or strongly agreed that it was easy and quick to recover from mistakes using the interface. P11 who disagreed thought it was hard to figure out a way

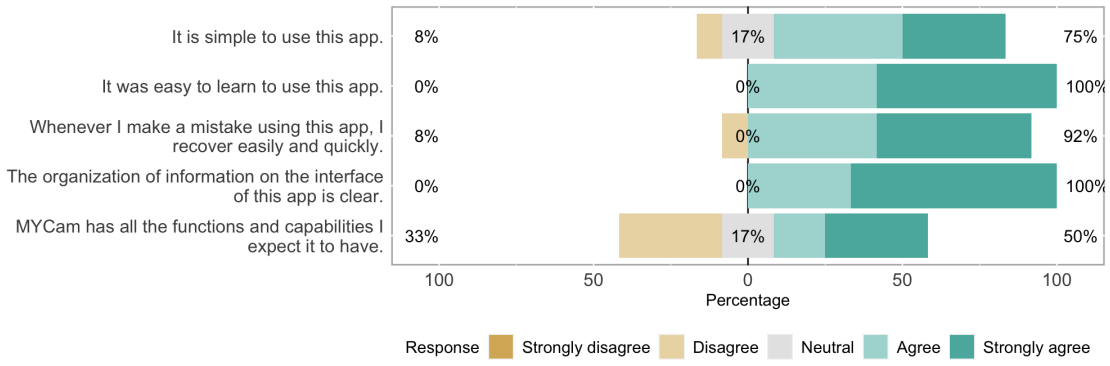

Figure 9.19: Participant responses to questions about their overall experience during the study.

to fix problems in the descriptors. She said *"If you're getting bad images, you need to take several sometimes as you're trying to figure out what angles and everything to use. Honestly, it's not quick, not really efficient."* We got a similar response from P10 when we asked participants if they agree that the TOR app has all functions and capabilities they expect it to have. P10 elaborated *"What does it mean? when it says it's cropped? [...] Like, if you get this feedback, what should you do? They don't know what to do."* Four participants disagreed that the app has all important functions and capabilities. They expected the app to provide: more detailed information of the snacks such as ingredients in recognition results (P2), better performance of object recognition (P5), an interface to replace a subset of training examples (P6), and feedback for fixing problems in descriptors (P10).

9.4 Discussion

9.4.1 Usability Issues of TOR

Through the evaluation of the app design, we observed that participants could easily understand and carry out the tasks of training an object recognition model, testing it, and managing the information of objects in their datasets. While the responses from the participants were also positive with regard to the user experience with the app, they also provided issues that provides useful insights in designing a TOR app in the future. One of the critical problem was that taking many photos for training (*i.e.*, 30 photos in this study) would be tedious for some blind users. Some possible solutions for this problem would be using computer vision techniques that require smaller number of photos (*e.g.*, one-shot learning [206]) or extracting frames from a short video. To achieve this, a future study is needed to find a good number of photos that can balance the performance of an object recognizer and the usability of a teachable interface.

While the descriptors just approximately estimated the attributes of photos in this study, the majority of the participants thought they are helpful. They also pointed out some ways to improve the interaction with the descriptors. For example, P8 suggested having an interface that filters out bad images based on descriptors or enables users to replace them instead of retaking all images. P10 mentioned a challenge in resolving problems in descriptors because they inform users of the problems in photos without providing ways to solve them. For example, when an object is cropped in a photo, participants did not get feedback on in which direction the camera should move. This

indicates that combining the descriptors with systems that provide audio/haptic feedback for blind photography (*e.g.*, [141, 146]) would make the descriptors more effective.

When it comes to the usefulness of a teachable interface, participants had different attitudes. From the participants' responses, we could find both participants who valued the possibility of recognizing personal items with the TOR app and who raised questions about the need of a teachable interface for object recognition. The participants with questions thought they would not need an object recognizer if they can train it because they already know about the object. Though there are some scenarios where identifying personal items can be useful (*e.g.*, scanning the surroundings to find it, distinguishing similar objects) and assistive apps in the market (*e.g.*, Seeing AI, LookTel) are deploying teachable interfaces in object recognizer, blind users may not have a clear motivation to use a teachable interface without instructions. Therefore, as an emerging technology for accessibility, a teachable interface would need to be incorporated with descriptions of real-world scenarios where blind people can use it.

9.4.2 Difference Between the User Study and Real Use Cases

Though we set up a controlled user study to simulate a real scenario of using the app, the study has a few limitations that makes difference from it. The limitations highlight the need of a future study with a deployment study that enables participants to use it in their environments. One of the limitations is the fact that participants were asked to wear smart glasses and communicate with the experimenter through a laptop computer in front of them. Though these devices were necessary for communication and data analysis,

they would limit the participants behavior such as walking around with the phone and finding a place for taking photos. For example, when participants wanted to vary the backgrounds in photos, they took pictures with different parts of a table as backgrounds. However, if they can move around outside the user study setup, they would be able to choose completely different locations for background variation. Another limitation is that all participants had to use the app on iPhone 8 instead of their own mobile devices. As all but one participant have used iPhone, most participants would be familiar with using iOS apps. However, the size of the device and position of the camera would affect the quality of photos taken without vision.

9.5 Conclusion

We designed and implemented a mobile TOR app for blind users. We aimed to resolve the known issues found in prior studies on interaction between TOR and blind users. The TOR app design was evaluated through a user study with blind participants. The user study also provided some patterns in training and testing an object recognition model through an analysis of feedback and photos from the participants. The responses to questions on usability of the app revealed that participants could easily train the app with descriptors, evaluate it with their test samples, and manage information of the objects during the study. However, participants also pointed out some difficulties such as taking many photos and resolving problems found in the descriptors. Moreover, we observed that the photos from participants had some issues such as little variation in training photos and cluttered background in test photos that would make the training and testing ineffective.

The findings from the user study provide insights and research problems to improve the usability of a TOR app for blind users.

Epilogue to Part II

In Part II of this thesis, the challenge of identifying image recognition errors and managing errors with TOR were characterized through crowdsourcing and controlled lab studies with blind and sighted participants. The studies investigated the blind users' experience in identifying errors in camera-based assistive tools and challenges in identifying object recognition errors. In the follow-up studies with TOR, we further looked into the interaction between blind users and teachable interface for object recognizer.

The study in Chapter 7 investigated the challenge in identifying errors from a pre-built camera-based assistive apps including object recognizer with blind users. It revealed that blind users identify errors based on the contexts such as the shape, size, and weight of the object. Like ASR errors, participants rarely had verified their the outputs from the camera-based assistive apps. On the other hand, while most blind people thought identifying ASR errors was not challenging, around half of the blind participants were aware of the difficulty of identifying image recognition errors. The results of error detection task showed that blind participants missed more than half of the object recognition errors.

The studies in Chapter 8, 9, and **??** characterized blind and sighted people's interactions with TOR. The analysis of their feedback and photos with web-based and mobile TOR

apps revealed that they tend to have some problems in their teaching strategies such as having little variation and cluttered photos that may cause errors in TOR. The responses of both blind and sighted participants in the user studies in Chapter 8 and **??** revealed that even if they observe the errors during the tests, many of them do not know what to do to resolve errors. However, some participants in Chapter **??** could figure out what to change in their strategies based on the descriptors of photos in our TOR app.

Part II answered the following research questions:

- **RQ7**: For what tasks and objects do blind users take photos? (The interview with blind participants in Chapter 7 showed that they mostly take photos for text recognition, video call, and object recognition.)

- **RQ8**: How did blind users identify the image recognition errors? (The most common way for blind people to identify image recognition errors reported during the interview in Chapter 7 was to decide the correctness for themselves based on the context (*e.g.*, surrounding texts of the recognized text, comparing the object recognition result with the shape, size, and texture of the object).

- **RQ9**: What are the blind users' accuracy of identifying the object recognition errors? (During the error identification task in Chapter 7, participants identified 49% of the image recognition errors.)

- **RQ10**: What are their strategies of identifying the errors? (Blind participants in Chapter 7 compared the recognition results with the expected objects based on the weight, texture, and shape in most cases. Some participants compared the

recognition results between trials. Some participants with low vision used the perceived colors and shapes to decide the correctness.)

- **RQ11**: What are non-experts' teaching and debugging strategies for a teachable object recognizer? (The study in Chapter 8 revealed both promising trends (*e.g.*, incorporating diversity in training examples) and misconceptions (*e.g.*, inconsistency between classes) in photos collected by non-experts.)

- **RQ12**: Do teaching strategies evolve through iteration? (The study in Chapter 8 showed that it did not evolve significantly because non-experts did not know what to change or did not want to change their strategies.)

- **RQ13**: How could descriptors be useful for avoiding errors due to their training examples? (Participants in the user study in Chapter 9 could learn what is important to consider to collect good training examples from the descriptors.)

- **RQ14**: What are blind users' teaching and debugging patterns? (The analysis of their photos in Chapter 9 showed similar trends found in Chapter 8 while they also had image framing problems.)

Chapter 11: Conclusions and Future Work

11.1 Summary of Contributions

This dissertation characterized the blind and sighted users' interactions with errors in speech and image recognition systems. As a conclusion, this chapter summarizes the key contributions of the thesis and presents directions for future research. Overall, the contributions of this thesis are related to speech recognition, object recognition systems, and machine teaching, and accessibility.

- Speech recognition systems: the basis of evaluating the accuracy of identifying ASR errors with the baseline accuracy of error identification; possibility of enabling users to identify the errors more accurately with manipulations of synthesized speech.
- Object recognition systems: understanding the challenges of using camera-based assistive apps; understanding non-experts' strategies for training and testing a teachable object recognizer; enabling blind users to understand and avoid errors by reviewing their training examples with descriptors.
- Machine teaching: identifying research problems in building teachable object recognizer for non-experts in machine learning and blind users.

11.2 Future Directions

My dissertation research explored the challenge of identifying and avoiding errors in speech and teachable object recognizers with blind and sighted users. With the findings and observations in my research, I highlight the following future directions to facilitate identifying, understanding, and correcting errors in AI-infused systems for blind and sighted people.

11.2.1 Interacting with Speech Recognition Errors

Enabling users to better identify ASR errors with audio-only interactions. As devices with no or very small visual displays (*e.g.*, smart speakers, wearable devices) where speech input is useful have been increasingly popular, audio-based interactions for identifying, understanding, and correcting errors can be more frequently used in various types of applications such as text editor, office tools, and social media apps. However, audio-only interaction for text entry is still an under-explored area. My previous study presented promising and simple manipulations that increased the accuracy of error identification (*i.e.*, adding pauses between words, using slower speech rates). However, even with the manipulations, the best average proportion of identified errors was around 0.70, which indicates room for more improvement with elaborate audio-only error identification support. That is, improvement is necessary to bring audio-only text input more in line with the accuracy that can be achieved with the visual text entry interface. One possibility is to explore audio techniques that are comparable to visually underlining words that the recognition system deems to be potentially incorrect.

User interface for correcting errors with audio-only interactions. Though my dissertation research focused on the challenge of identifying ASR errors through synthetic speech, a system using speech input needs to incorporate user interfaces not only for error identification but also for error understanding and correction. A possible approach for enabling users to go through error identification, understanding, and correction in a non-visual context would be a dialogue-based interface that allows users to indicate errors and re-speak for the misrecognized words through a conversation with the system. We see that some of the strategies for pointing to errors found in my previous work would be leveraged to develop dialogue-based interfaces in future work.

11.2.2 Interacting with Error-Prone Image Recognition

Effective teachable interfaces for real-world scenarios. While many parameters in teachable object recognizers can be explored in the future such as incremental model learning, extreme illumination changes, and video versus images in training, a critical remaining issue is a scalability over a long period of time. Although my thesis and prior studies have shown success for moderately sized datasets (*e.g.*, fewer than 20 objects, around 30 photos per object), the number of objects would increase over time to hundreds or thousands in real use cases. The scalability problem affects both performance and usability. As a dataset include more objects, a teachable object recognizer inevitably makes more errors because the object recognition task becomes harder. Moreover, like machine learning practitioners put much effort in controlling a large dataset from various perspectives (*e.g.*, fairness, diversity, consistent distribution of data in training and real-

world data), it will be a tricky task for end-users to manage their datasets with many classes collected over time. Therefore, future studies are needed to facilitate data management and model evaluation processes with scalability for end-users.

Descriptors in other teachable assistive applications. My thesis explored the challenge of understanding and avoiding errors in teachable object recognizers, where 'learning to train' is deemed as one of the main challenges among blind users. As a way to resolve this challenge, we presented descriptors that allowed blind users to understand the important attributes of a training dataset and to evaluate their training examples quantitatively. While my thesis focused on a teachable object recognizer, this challenge is common among teachable assistive applications where users are required to build their own datasets. We see how the underlying methods for extracting meaningful descriptors can be adopted for other teachable assistive technologies and user groups such as teachable sound detectors for Deaf/deaf and hard of hearing people.

Bibliography

[1] Jonggi Hong, Kyungjun Lee, June Xu, and Hernisa Kacorri. Exploring machine teaching for object recognition with the crowd. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19, New York, NY, USA, 2019. Association for Computing Machinery.

[2] Utkarsh Dwivedi, Jaina Gandhi, Raj Parikh, Merijke Coenraad, Elizabeth Bonsignore, and Hernisa Kacorri. Exploring machine teaching with children. In *2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 2021.

[3] Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy? – a comprehensive study on the robustness of 18 deep image classification models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[4] Xiaojin Zhu, Adish Singla, Sandra Zilles, and Anna N. Rafferty. An overview of machine teaching. *CoRR*, abs/1801.05927, 2018.

[5] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[6] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, 2015.

[7] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke. The microsoft 2017 conversational speech recognition system. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5934–5938, 2018.

[8] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 4780–4789, 2019.

[9] Yingbo Zhou, Caiming Xiong, and Richard Socher. Improving end-to-end speech recognition with policy learning. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5819–5823. IEEE, 2018.

[10] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, and et al. Guidelines for human-ai interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, New York, NY, USA, 2019. Association for Computing Machinery.

[11] Andrew Begel, John Tang, Sean Andrist, Michael Barnett, Tony Carbary, Piali Choudhury, Edward Cutrell, Alberto Fung, Sasa Junuzovic, Daniel McDuff, Kael Rowan, Shibashankar Sahoo, Jennifer Frances Waldern, Jessica Wolk, Hui Zheng, and Annuska Zolyomi. Lessons learned in designing ai for autistic adults. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '20, New York, NY, USA, 2020. Association for Computing Machinery.

[12] Matthew A Fox, Carl J Aschkenasi, and Arjun Kalyanpur. Voice recognition is here comma like it or not period. *The Indian journal of radiology & imaging*, 23(3):191, 2013.

[13] Shiri Azenkot and Nicole B. Lee. Exploring the use of speech input by blind people on mobile devices. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '13, New York, NY, USA, 2013. Association for Computing Machinery.

[14] Christine A Halverson, Daniel B Horn, Clare-Marie Karat, and John Karat. The beauty of errors: Patterns of error correction in desktop speech systems. In *INTERACT*, volume 99, pages 1–8, 1999.

[15] Clare-Marie Karat, Christine Halverson, Daniel Horn, and John Karat. Patterns of entry and correction in large vocabulary continuous speech recognition systems. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 568–575, 1999.

[16] Ben Shneiderman. The limits of speech recognition. *Communications of the ACM*, 43(9):63–65, 2000.

[17] Michael A Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4845–4854, 2019.

[18] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[19] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016.

[20] Hernisa Kacorri, Kris M. Kitani, Jeffrey P. Bigham, and Chieko Asakawa. 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, page 5839–5849, New York, NY, USA, 2017. Association for Computing Machinery.

[21] Carrie J. Cai, Jonas Jongejan, and Jess Holbrook. The effects of example-based explanations in a machine learning interface. IUI '19, page 258–262, New York, NY, USA, 2019. Association for Computing Machinery.

[22] Andrew Fowler, Brian Roark, Umut Orhan, Deniz Erdogmus, and Melanie Fried-Oken. Improved inference and autotyping in eeg-based bci typing systems. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 1–8, 2013.

[23] Heidi Horstmann Koester and Simon P Levine. Validation of a keystroke-level model for a text entry system used by people with disabilities. In *Proceedings of the first annual ACM conference on Assistive technologies*, pages 115–122, 1994.

[24] Hernisa Kacorri and Matt Huenerfauth. Continuous profile models in asl syntactic facial expression synthesis. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2084–2093, 2016.

[25] Cole Gleason, Anhong Guo, Gierad Laput, Kris Kitani, and Jeffrey P Bigham. Vizmap: Accessible visual information through crowdsourced map reconstruction. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 273–274, 2016.

[26] Daisuke Sato, Uran Oh, Kakuya Naito, Hironobu Takagi, Kris Kitani, and Chieko Asakawa. Navcog3: An evaluation of a smartphone-based blind indoor navigation assistant with semantic features in a large-scale environment. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 270–279, 2017.

[27] Tom Kontogiannis. User strategies in recovering from errors in man–machine systems. *Safety Science*, 32(1):49–68, 1999.

[28] Tom Kontogiannis and Stathis Malakis. A proactive approach to human error detection and identification in aviation and air traffic control. *Safety Science*, 47(5):693 – 706, 2009.

[29] Abigail J Sellen. Detection of everyday errors. *Applied Psychology*, 43(4):475–498, 1994.

[30] Marie-Luce Bourguet. Towards a taxonomy of error-handling strategies in recognition-based multi-modal human–computer interfaces. *Signal Processing*, 86(12):3625–3643, 2006.

[31] Bernhard Suhm, Brad Myers, and Alex Waibel. Model-based and empirical evaluation of multimodal interactive error correction. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 584–591, 1999.

[32] Takeo Igarashi, Satoshi Matsuoka, Sachiko Kawachiya, and Hidehiko Tanaka. Interactive beautification: A technique for rapid geometric design. In *ACM SIGGRAPH 2007 courses*, pages 18–es. 2007.

[33] Haley MacLeod, Cynthia L Bennett, Meredith Ringel Morris, and Edward Cutrell. Understanding blind people's experiences with computer-generated captions of social media images. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 5988–5999, 2017.

[34] Rafal Kocielnik, Saleema Amershi, and Paul N. Bennett. Will you accept an imperfect ai? exploring designs for adjusting end-user expectations of ai systems. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–14, New York, NY, USA, 2019. Association for Computing Machinery.

[35] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250, 2000.

[36] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th international conference on intelligent user interfaces*, pages 126–137, 2015.

[37] Brian Y Lim and Anind K Dey. Assessing demand for intelligibility in context-aware applications. In *Proceedings of the 11th international conference on Ubiquitous computing*, pages 195–204, 2009.

[38] Emilee Rader, Kelley Cotter, and Janghee Cho. Explanations as mechanisms for supporting algorithmic transparency. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–13, 2018.

[39] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

[40] Daniel S Weld and Gagan Bansal. The challenge of crafting intelligible intelligence. *Communications of the ACM*, 62(6):70–79, 2019.

[41] René F Kizilcec. How much information? effects of transparency on trust in an algorithmic interface. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2390–2395, 2016.

[42] Todd Kulesza, Simone Stumpf, Margaret Burnett, and Irwin Kwan. Tell me more? the effects of mental model soundness on personalizing an intelligent agent. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1–10, 2012.

[43] Ángel Alexander Cabrera, Fred Hohman, Jason Lin, and Duen Horng Chau. Interactive classification for deep learning interpretation. *arXiv preprint arXiv:1806.05660*, 2018.

[44] Pearl Pu and Li Chen. Trust building with explanation interfaces. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 93–100, 2006.

[45] Rahhal Errattahi, Asmaa El Hannani, Thomas Hain, and Hassan Ouahmane. Towards a generic approach for automatic speech recognition error detection and classification. In *2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pages 1–6. IEEE, 2018.

[46] Manaswi Saha, Alexander J Fiannaca, Melanie Kneisel, Edward Cutrell, and Meredith Ringel Morris. Closing the gap: Designing for the last-few-meters wayfinding problem for people with visual impairments. In *The 21st international acm sigaccess conference on computers and accessibility*, pages 222–235, 2019.

[47] Robin N Brewer and Vaishnav Kameswaran. Understanding the power of control in autonomous vehicles for people with vision impairment. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 185–197, 2018.

[48] Julian Brinkley, Brianna Posadas, Julia Woodward, and Juan E Gilbert. Opinions and preferences of blind and low vision consumers regarding self-driving vehicles: Results of focus group discussions. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 290–299, 2017.

[49] Ali Abdolrahmani, William Easley, Michele Williams, Stacy Branham, and Amy Hurst. Embracing errors: Examining how context of use impacts blind individuals' acceptance of navigation aid errors. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4158–4169, 2017.

[50] Kristina Höök. Steps to take before intelligent user interfaces become real. *Interacting with computers*, 12(4):409–426, 2000.

[51] Donald A Norman. How might people interact with agents. *Communications of the ACM*, 37(7):68–71, 1994.

[52] Tomoko Hashida, Kohei Nishimura, and Takeshi Naemura. Hand-rewriting: automatic rewriting similar to natural handwriting. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces*, pages 153–162, 2012.

[53] David Dearman, Amy Karlson, Brian Meyers, and Ben Bederson. Multi-modal text entry and selection on a mobile device. In *Proceedings of Graphics Interface 2010*, pages 19–26. 2010.

[54] Clive Frankish, Dylan Jones, and Kevin Hapeshi. Decline in accuracy of automatic speech recognition as a function of time on task: fatigue or voice drift? *International Journal of Man-Machine Studies*, 36(6):797–816, 1992.

[55] Sharon Oviatt and Robert VanGent. Error resolution during multimodal human-computer interaction. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, volume 1, pages 204–207. IEEE, 1996.

[56] Anja Kintsch and Rogerio DePaula. A framework for the adoption of assistive technology. *SWAAAC 2002: Supporting learning through assistive technology*, pages 1–10, 2002.

[57] Betsy Phillips and Hongxin Zhao. Predictors of assistive technology abandonment. *Assistive technology*, 5(1):36–45, 1993.

[58] Shaun K Kane, Chandrika Jayant, Jacob O Wobbrock, and Richard E Ladner. Freedom to roam: a study of mobile device adoption and accessibility for people with visual and motor disabilities. In *Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility*, pages 115–122, 2009.

[59] Sherry Ruan, Jacob O Wobbrock, Kenny Liou, Andrew Ng, and James Landay. Speech is 3x faster than typing for english and mandarin text entry on mobile devices. *arXiv preprint arXiv:1608.07323*, 2016.

[60] Hanlu Ye, Meethu Malu, Uran Oh, and Leah Findlater. Current and future mobile and wearable device use by people with visual impairments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3123–3132, 2014.

[61] Hui Jiang. Confidence measures for speech recognition: A survey. *Speech communication*, 45(4):455–470, 2005.

[62] W Feng. Using handwriting and gesture recognition to correct speech recognition errors. *Urbana*, 51:61801, 1994.

[63] Arnout RH Fischer, Kathleen J Price, and Andrew Sears. Speech-based text entry for mobile handheld devices: an analysis of efficacy and error correction techniques for server-based solutions. *International Journal of Human-Computer Interaction*, 19(3):279–304, 2005.

[64] Kazuki Fujiwara. Error correction of speech recognition by custom phonetic alphabet input for ultra-small devices. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 104–109, 2016.

[65] Vidyashree Kanabur, Sunil S Harakannanavar, and Dattaprasad Torse. An extensive review of feature extraction techniques, challenges and trends in automatic speech recognition. *International Journal of Image, Graphics and Signal Processing*, 10(5):1, 2019.

[66] Arul Valiyavalappil Haridas, Ramalatha Marimuthu, and Vaazi Gangadharan Sivakumar. A critical review and analysis on techniques of speech recognition: The road ahead. *International Journal of Knowledge-Based and Intelligent Engineering Systems*, 22(1):39–57, 2018.

[67] D Walker. The sri speech understanding system. *IEEE transactions on acoustics, speech, and signal processing*, 23(5):397–416, 1975.

[68] Parabattina Bhagath and Pradip K Das. Acoustic phonetic approach for speech recognition: A review. *Language*, 77:93, 2004.

[69] Zhehuai Chen, Mahaveer Jain, Yongqiang Wang, Michael L Seltzer, and Christian Fuegen. End-to-end contextual speech recognition using class language models and a token passing decoder. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6186–6190. IEEE, 2019.

[70] Sharon Goldwater, Dan Jurafsky, and Christopher D Manning. Which words are hard to recognize? prosodic, lexical, and disfluency factors that increase speech recognition error rates. *Speech Communication*, 52(3):181–200, 2010.

[71] Rahhal Errattahi, Asmaa El Hannani, and Hassan Ouahmane. Automatic speech recognition errors detection and correction: A review. *Procedia Computer Science*, 128:32–37, 2018.

[72] Larwan Berke, Christopher Caulfield, and Matt Huenerfauth. Deaf and hard-of-hearing perspectives on imperfect automatic speech recognition for captioning one-on-one meetings. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 155–164, 2017.

[73] Yik-Cheung Tam, Yun Lei, Jing Zheng, and Wen Wang. Asr error detection using recurrent neural network language model and complementary asr. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2312–2316. IEEE, 2014.

[74] Sahar Ghannay, Nathalie Camelin, and Yannick Esteve. Which asr errors are hard to detect. In *Errors by Humans and Machines in Multimedia, Multimodal and Multilingual Data Processing (ERRARE 2015) Workshop, Sinaia, Romania*, pages 11–13, 2015.

[75] Sahar Ghannay, Yannick Esteve, and Nathalie Camelin. Word embeddings combination and neural networks for robustness in asr error detection. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 1671–1675. IEEE, 2015.

[76] David Huggins-Daines and Alexander Rudnicky. Interactive asr error correction for touchscreen devices. In *Proceedings of the ACL-08: HLT Demo Session*, pages 17–19, 2008.

[77] Yuan Liang, Koji Iwano, and Koichi Shinoda. Simple gesture-based error correction interface for smartphone speech recognition. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[78] Jun Ogata and Masataka Goto. Speech repair: quick error correction just by using selection operation for speech input interfaces. In *Ninth European Conference on Speech Communication and Technology*, 2005.

[79] Lijuan Wang, Tao Hu, Peng Liu, and Frank K Soong. Efficient handwriting correction of speech recognition errors with template constrained posterior (tcp). In *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[80] Junhwi Choi, Kyungduk Kim, Sungjin Lee, Seokhwan Kim, Donghyeon Lee, Injae Lee, and Gary Geunbae Lee. Seamless error correction interface for voice word processor. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4973–4976. IEEE, 2012.

[81] Vikas Ashok, Yevgen Borodin, Yury Puzis, and IV Ramakrishnan. Capti-speak: a speech-enabled web screen reader. In *Proceedings of the 12th Web for All Conference*, pages 1–10, 2015.

[82] Yevgen Borodin, Jalal Mahmud, IV Ramakrishnan, and Amanda Stent. The hearsay non-visual web browser. In *Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A)*, pages 128–129, 2007.

[83] Alisha Pradhan, Kanika Mehta, and Leah Findlater. " accessibility came by accident" use of voice-controlled intelligent personal assistants by people with disabilities. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2018.

[84] Yu Zhong, TV Raman, Casey Burkhardt, Fadi Biadsy, and Jeffrey P Bigham. Justspeak: enabling universal voice control on android. In *Proceedings of the 11th Web for All Conference*, pages 1–4, 2014.

[85] Jeff Bilmes, Xiao Li, Jonathan Malkin, Kelley Kilanski, Richard Wright, Katrin Kirchhoff, Amarnag Subramanya, Susumu Harada, James Landay, Patricia Dowden, et al. The vocal joystick: A voice-based human-computer interface for individuals with motor impairments. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 995–1002, 2005.

[86] Eric Corbett and Astrid Weber. What can i say? addressing user experience challenges of a mobile voice user interface for accessibility. In *Proceedings of the*

18th international conference on human-computer interaction with mobile devices and services, pages 72–82, 2016.

[87] Susumu Harada, James A Landay, Jonathan Malkin, Xiao Li, and Jeff A Bilmes. The vocal joystick: evaluation of voice-based cursor control techniques for assistive technology. *Disability and Rehabilitation: Assistive Technology*, 3(1-2):22–34, 2008.

[88] Bill Manaris, Renée McCauley, and Valanne MacGyvers. An intelligent interface for keyboard and mouse control. In *Proc. 14th Int'l Florida AI Research Symposium (FLAIRS-01)*, pages 182–188. Citeseer, 2001.

[89] Yoshiyuki Mihara, Etsuya Shibayama, and Shin Takahashi. The migratory cursor: accurate speech-based cursor movement by moving multiple ghost cursors using non-verbal vocalizations. In *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 76–83, 2005.

[90] Stephen Cox, Michael Lincoln, Judy Tryggvason, Melanie Nakisa, Mark Wells, Marcus Tutt, and Sanja Abbott. Tessa, a system to aid communication with deaf people. In *Proceedings of the fifth international ACM conference on Assistive technologies*, pages 205–212, 2002.

[91] Thomas Pellegrini, Lionel Fontan, Julie Mauclair, Jérôme Farinas, Charlotte Alazard-Guiu, Marina Robert, and Peggy Gatignol. Automatic assessment of speech capability loss in disordered speech. *ACM Transactions on Accessible Computing (TACCESS)*, 6(3):1–14, 2015.

[92] Oscar Saz, Shou-Chun Yin, Eduardo Lleida, Richard Rose, Carlos Vaquero, and William R Rodríguez. Tools and technologies for computer-aided speech and language therapy. *Speech Communication*, 51(10):948–967, 2009.

[93] Mark S Hawley, Stuart P Cunningham, Phil D Green, Pam Enderby, Rebecca Palmer, Siddharth Sehgal, and Peter O'Neill. A voice-input voice-output communication aid for people with severe speech impairment. *IEEE Transactions on neural systems and rehabilitation engineering*, 21(1):23–31, 2012.

[94] Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y Lim, and Mohan Kankanhalli. Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–18, 2018.

[95] Konstantinos Papadopoulos and Eleni Koustriava. Comprehension of synthetic and natural speech: Differences among sighted and visually impaired young adults. *Enabling Access for Persons with Visual Impairment*, 147:149–153, 2015.

[96] Amanda Stent, Ann Syrdal, and Taniya Mishra. On the intelligibility of fast synthesized speech for individuals with early-onset blindness. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, pages 211–218, 2011.

[97] Danielle Bragg, Cynthia Bennett, Katharina Reinecke, and Richard Ladner. A large inclusive study of human listening rates. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2018.

[98] Anja Moos and Jürgen Trouvain. Comprehension of ultra-fast speech–blind vs.'normally hearing'persons. In *Proceedings of the 16th International Congress of Phonetic Sciences*, volume 1, pages 677–680. Saarland University Saarbrücken, Germany, 2007.

[99] E Colin Cherry. Some experiments on the recognition of speech, with one and with two ears. *The Journal of the acoustical society of America*, 25(5):975–979, 1953.

[100] João Guerreiro and Daniel Gonçalves. Scanning for digital content: How blind and sighted people perceive concurrent speech. *ACM Transactions on Accessible Computing (TACCESS)*, 8(1):1–28, 2016.

[101] Ann R Bradlow and Jennifer A Alexander. Semantic and phonetic enhancements for speech-in-noise recognition by native and non-native listeners. *The Journal of the Acoustical Society of America*, 121(4):2339–2349, 2007.

[102] Brenda Sutton, Julia King, Karen Hux, and David Beukelman. Younger and older adults' rate performance when listening to synthetic speech. *Augmentative and Alternative Communication*, 11(3):147–153, 1995.

[103] Kelly Caine. Local standards for sample size at chi. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 981–992, 2016.

[104] Zhirong Wang, Tanja Schultz, and Alex Waibel. Comparison of acoustic model adaptation techniques on non-native speech. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 1, pages I–I. IEEE, 2003.

[105] Mary LaLomia. User acceptance of handwritten recognition accuracy. In *Conference companion on Human factors in computing systems*, pages 107–108, 1994.

[106] Antti Oulasvirta, Sakari Tamminen, Virpi Roto, and Jaana Kuorelahti. Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile hci. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 919–928, 2005.

[107] Graham Upton and Ian Cook. *Understanding statistics*. Oxford University Press, 1996.

[108] Larry D Rosen, Kelly Whaling, L Mark Carrier, Nancy A Cheever, and Jeffrey Rokkum. The media and technology usage and attitudes scale: An empirical investigation. *Computers in human behavior*, 29(6):2501–2511, 2013.

[109] Sorrel Brown. Likert scale examples for surveys. *ANR Program evaluation, Iowa State University, USA*, 2010.

[110] Marcelo Philip. Technology seeks to preserve fading skill: Braille literacy. *AP Financial*, 2017.

[111] Keith Vertanen and Per Ola Kristensson. Complementing text entry evaluations with a composition task. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 21(2):1–33, 2014.

[112] Michael Gonchar. 650 prompts for narrative and personal writing. *New York Times*, 20, 2016.

[113] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101, 2006.

[114] William R Revelle. psych: Procedures for personality and psychological research. 2017.

[115] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[116] Sara E McBride, Wendy A Rogers, and Arthur D Fisk. Understanding human management of automation errors. *Theoretical issues in ergonomics science*, 15(6):545–577, 2014.

[117] Xiaodong He, Li Deng, and Alex Acero. Why word error rate is not a good metric for speech recognizer training for the speech translation task? In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5632–5635. IEEE, 2011.

[118] Sushant Kafle and Matt Huenerfauth. Evaluating the usability of automatically generated captions for people who are deaf or hard of hearing. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 165–174, 2017.

[119] Danielle Bragg, Nicholas Huynh, and Richard E Ladner. A personalizable mobile sound detector app design for deaf and hard-of-hearing users. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 3–13, 2016.

[120] Xin Zhang, Yee-Hong Yang, Zhiguang Han, Hui Wang, and Chao Gao. Object class detection: A survey. *ACM Computing Surveys (CSUR)*, 46(1):1–53, 2013.

[121] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International journal of computer vision*, 128(2):261–318, 2020.

[122] Jinqiang Bai, Dijun Liu, Guobin Su, and Zhongliang Fu. A cloud and vision-based navigation system used for blind people. In *Proceedings of the 2017 International Conference on Artificial Intelligence, Automation and Control Technologies*, pages 1–6, 2017.

[123] YingLi Tian, Xiaodong Yang, Chucai Yi, and Aries Arditi. Toward a computer vision-based wayfinding aid for blind persons to access unfamiliar indoor environments. *Machine vision and applications*, 24(3):521–535, 2013.

[124] SeeingAI. An app for visually impaired people that narrates the world around you, 2017.

[125] Aipoly. Vision through artificial intelligence, 2016.

[126] Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya. Captioning images taken by people who are blind. In *European Conference on Computer Vision*, pages 417–434. Springer, 2020.

[127] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[128] Alexander Andreopoulos and John K Tsotsos. 50 years of object recognition: Directions forward. *Computer vision and image understanding*, 117(8):827–891, 2013.

[129] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[130] Meredith Ringel Morris. Ai and accessibility. *Communications of the ACM*, 63(6):35–37, 2020.

[131] Rabia Jafri, Syed Abid Ali, Hamid R Arabnia, and Shameem Fatima. Computer vision-based object recognition for the visually impaired in an indoors environment: a survey. *The Visual Computer*, 30(11):1197–1222, 2014.

[132] Alejandro Reyes-Amaro, Yanet Fadraga-González, Oscar Luis Vera-Pérez, Elizabeth Domínguez-Campillo, Jenny Nodarse-Ravelo, Alejandro Mesejo-Chiong, Biel Moyà-Alcover, and Antoni Jaume-i Capó. Rehabilitation of patients with motor disabilities using computer vision based techniques. *Journal of accessibility and design for all*, 2(1):62–70, 2012.

[133] Taha Khan, Dag Nyholm, Jerker Westin, and Mark Dougherty. A computer vision framework for finger-tapping evaluation in parkinson's disease. *Artificial intelligence in medicine*, 60(1):27–40, 2014.

[134] Hairong Jiang, Ting Zhang, Juan P Wachs, and Bradley S Duerstock. Enhanced control of a wheelchair-mounted robotic manipulator using 3-d vision and multimodal interaction. *Computer Vision and Image Understanding*, 149:21–31, 2016.

[135] Cristina Manresa-Yee, Javier Varona, Francisco J Perales, and Iosune Salinas. Design recommendations for camera-based head-controlled interfaces that replace the mouse for motion-impaired users. *Universal access in the information society*, 13(4):471–482, 2014.

[136] Kathleen Campbell, Kimberly LH Carpenter, Jordan Hashemi, Steven Espinosa, Samuel Marsan, Jana Schaich Borg, Zhuoqing Chang, Qiang Qiu, Saritha Vermeer, Elizabeth Adler, et al. Computer vision analysis captures atypical attention in toddlers with autism. *Autism*, 23(3):619–628, 2019.

[137] Jordan Hashemi, Mariano Tepper, Thiago Vallin Spina, Amy Esler, Vassilios Morellas, Nikolaos Papanikolopoulos, Helen Egger, Geraldine Dawson, and Guillermo Sapiro. Computer vision tools for low-cost and noninvasive measurement of autism-related behaviors in infants. *Autism research and treatment*, 2014, 2014.

[138] Ruxandra Tapu, Bogdan Mocanu, and Titus Zaharia. Deep-hear: A multimodal subtitle positioning system dedicated to deaf and hearing-impaired people. *IEEE Access*, 7:88150–88162, 2019.

[139] Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, Annelies Braffort, Naomi Caselli, Matt Huenerfauth, Hernisa Kacorri, Tessa Verhoef, et al. Sign language recognition, generation, and translation: An interdisciplinary perspective. In *The 21st international ACM SIGACCESS conference on computers and accessibility*, pages 16–31, 2019.

[140] Kyungjun Lee and Hernisa Kacorri. Hands holding clues for object recognition in teachable machines. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

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

[142] Chandrika Jayant, Hanjie Ji, Samuel White, and Jeffrey P Bigham. Supporting blind photography. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, pages 203–210, 2011.

[143] TapTapSee. Mobile camera application designed specifically for the blind and visually impaired ios users, 2016.

[144] Envision AI. Enabling vision for the blind., 2018.

[145] Aira. Your life, your schedule, right now., 2017.

[146] Kyungjun Lee, Jonggi Hong, Simone Pimento, Ebrima Jarjue, and Hernisa Kacorri. Revisiting blind photography in the context of teachable object recognizers. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*, pages 83–95, 2019.

[147] E. Johns, O. M. Aodha, and G. J. Brostow. Becoming the expert - interactive multi-class machine teaching. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2616–2624, June 2015.

[148] Andrea L. Thomaz and Cynthia Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172(6):716 – 737, 2008.

[149] Rüdiger Dillmann. Teaching and learning of robot tasks via observation of human performance. *Robotics and Autonomous Systems*, 47(2):109 – 116, 2004. Robot Learning from Demonstration.

[150] Rupal Patel and Deb Roy. Teachable interfaces for individuals with dysarthric speech and severe physical disabilities. In *Proceedings of the AAAI Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 40–47. Citeseer, 1998.

[151] Tom Hitron, Yoav Orlev, Iddo Wald, Ariel Shamir, Hadas Erel, and Oren Zuckerman. Can children understand machine learning concepts? the effect of uncovering black boxes. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, New York, NY, USA, 2019. Association for Computing Machinery.

[152] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, Oct 2010.

[153] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2019.

[154] Hernisa Kacorri. Teachable machines for accessibility. *SIGACCESS Access. Comput.*, (119):10–18, November 2017.

[155] Rebecca Fiebrink, Perry R. Cook, and Dan Trueman. Human model evaluation in interactive supervised learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, page 147–156, New York, NY, USA, 2011. Association for Computing Machinery.

[156] I. I. Itauma, H. Kivrak, and H. Kose. Gesture imitation using machine learning techniques. In *2012 20th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4, April 2012.

[157] Abigail Zimmermann-Niefield, Makenna Turner, Bridget Murphy, Shaun K. Kane, and R. Benjamin Shapiro. Youth learning machine learning through building models of athletic moves. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children*, IDC '19, page 121–132, New York, NY, USA, 2019. Association for Computing Machinery.

[158] Thomas J Palmeri and Isabel Gauthier. Visual object understanding. *Nature Reviews Neuroscience*, 5(4):291, 2004.

[159] Jerry Alan Fails and Dan R. Olsen. Interactive machine learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, page 39–45, New York, NY, USA, 2003. Association for Computing Machinery.

[160] Kyungjun Lee, Daisuke Sato, Saki Asakawa, Hernisa Kacorri, and Chieko Asakawa. Pedestrian detection with wearable cameras for the blind: A two-way perspective. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020.

[161] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, June 2016.

[162] BeMyEyes. Lend you eyes to the blind, 2016.

[163] Donghee Shin. The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable ai. *International Journal of Human-Computer Studies*, 146:102551, 2021.

[164] Eric S Vorm. Assessing demand for transparency in intelligent systems using machine learning. In *2018 Innovations in Intelligent Systems and Applications (INISTA)*, pages 1–7. IEEE, 2018.

[165] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. *ProPublica*, 2016.

[166] Solon Barocas and Andrew D Selbst. Big data's disparate impact. *Cal. L. Rev.*, 104:671, 2016.

[167] danah boyd and Kate Crawford. Critical questions for big data. *Information, Communication & Society*, 15(5):662–679, 2012.

[168] Alex Campolo, Madelyn Sanfilippo, Meredith Whittaker, and Kate Crawford. Ai now 2017 report. *AI Now Institute at New York University*, 2017.

[169] Lucy A Suchman. *Plans and situated actions: The problem of human-machine communication*. Cambridge university press, 1987.

[170] William R. Swartout. Xplain: a system for creating and explaining expert consulting programs. *Artificial Intelligence*, 21(3):285 – 325, 1983.

[171] Ben Shneiderman and Pattie Maes. Direct manipulation vs. interface agents. *Interactions*, 4(6):42–61, November 1997.

[172] Nicholas Diakopoulos. Algorithmic accountability reporting: On the investigation of black boxes. 2014.

[173] Danding Wang, Qian Yang, Ashraf Abdul, and Brian Y. Lim. Designing theory-driven user-centric explainable ai. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, New York, NY, USA, 2019. Association for Computing Machinery.

[174] Daniel S Weld and Gagan Bansal. Intelligible artificial intelligence. *arXiv preprint arXiv:1803.04263*, 2018.

[175] David Gunning. Explainable artificial intelligence (xai), 2017.

[176] European Commision. European union general data protection regulation (gdpr), 2016.

[177] US Congress. S.1108 - algorithmic accountability act of 2019, 2019.

[178] Patrice Y. Simard, Saleema Amershi, David Maxwell Chickering, Alicia Edelman Pelton, Soroush Ghorashi, Christopher Meek, Gonzalo Ramos, Jina Suh, Johan Verwey, Mo Wang, and John Wernsing. Machine teaching: A new paradigm for building machine learning systems. *CoRR*, abs/1707.06742, 2017.

[179] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, and et al. Guidelines for human-ai interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, New York, NY, USA, 2019. Association for Computing Machinery.

[180] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105–120, Dec. 2014.

[181] Google Creative Lab. Teachable machine, 2017.

[182] Chien-Ju Ho, Aleksandrs Slivkins, Siddharth Suri, and Jennifer Wortman Vaughan. Incentivizing high quality crowdwork. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, page 419–429, Republic and Canton of Geneva, CHE, 2015. International World Wide Web Conferences Steering Committee.

[183] Flask API. Browsable web apis for flask, 2010.

[184] Kotaro Hara, Abigail Adams, Kristy Milland, Saiph Savage, Chris Callison-Burch, and Jeffrey P. Bigham. A data-driven analysis of workers' earnings on amazon mechanical turk. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 449:1–449:14, New York, NY, USA, 2018. ACM.

[185] Z. Gong, P. Zhong, and W. Hu. Diversity in machine learning. *IEEE Access*, 7:64323–64350, 2019.

[186] Arvind Narayanan. Fat* tutorial: 21 fairness definitions and their politics. *New York, NY, USA*, 2018.

[187] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning, 2019.

[188] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, page 214–226, New York, NY, USA, 2012. Association for Computing Machinery.

[189] Faisal Khan, Bilge Mutlu, and Jerry Zhu. How do humans teach: On curriculum learning and teaching dimension. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1449–1457. Curran Associates, Inc., 2011.

[190] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA, 2009. Association for Computing Machinery.

[191] Y. J. Lee and K. Grauman. Learning the easy things first: Self-paced visual category discovery. In *CVPR 2011*, pages 1721–1728, June 2011.

[192] Minjie Cai, Kris Kitani, and Yoichi Sato. Understanding hand-object manipulation by modeling the contextual relationship between actions, grasp types and object attributes, 2018.

[193] Qian Yang, Jina Suh, Nan-Chen Chen, and Gonzalo Ramos. Grounding interactive machine learning tool design in how non-experts actually build models. In *Proceedings of the 2018 Designing Interactive Systems Conference*, DIS '18, page 573–584, New York, NY, USA, 2018. Association for Computing Machinery.

[194] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, December 1974.

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

[196] Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. Trick me if you can: Human-in-the-loop generation of adversarial question answering examples. *Transactions of the Association for Computational Linguistics*, 7(0):387–401, 2019.

[197] Pierre Stock and Moustapha Cisse. Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[198] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *CoRR*, abs/1708.08296, 2017.

[199] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2014.

[200] Guangxiao Zhang, Zhuolin Jiang, and Larry S Davis. Online semi-supervised discriminative dictionary learning for sparse representation. In *Asian conference on computer vision*, pages 259–273. Springer, 2012.

[201] Jonggi Hong, Christine Vaing, Hernisa Kacorri, and Leah Findlater. Reviewing speech input with audio: Differences between blind and sighted users. *ACM Trans. Access. Comput.*, 13(1), April 2020.

[202] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[203] Joan Sosa-García and Francesca Odone. "hands on" visual recognition for visually impaired users. *ACM Transactions on Accessible Computing (TACCESS)*, 10(3):1–30, 2017.

[204] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

[205] A. Groce, T. Kulesza, C. Zhang, S. Shamasunder, M. Burnett, W. Wong, S. Stumpf, S. Das, A. Shinsel, F. Bice, and K. McIntosh. You are the only possible oracle: Effective test selection for end users of interactive machine learning systems. *IEEE Transactions on Software Engineering*, 40(3):307–323, March 2014.

[206] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3630–3638. Curran Associates, Inc., 2016.