Department of Computer Science
George Mason University   Technical Reports

4400 University Drive MS#4A5
Fairfax, VA 22030-4444 USA
http://cs.gmu.edu/   703-993-1530

HoloAAC: A Mixed Reality AAC Application for People with Expressive Language Difficulties

Liuchuan Yu
lyu20@gmu.edu

Huining Feng
hfeng2@gmu.edu

Rawan Alghofaili
ralghofa@gmu.edu

Boyoung Byun
cbyun2@gmu.edu

Tiffany O'Neal
toneal@gmu.edu

Swati Rampalli
rampa009@umn.edu

Yoosun Chung
ychung3@gmu.edu

Vivian Genaro Motti
vmotti@gmu.edu

Lap-Fai Yu
craigyu@gmu.edu

Technical Report GMU-CS-TR-2023-2

Abstract

We present a novel AAC application, HoloAAC, based on mixed reality that helps people with expressive language difficulties communicate in grocery shopping scenarios via a mixed reality device. A user, who has difficulty in speaking, can easily convey their intention by pressing a few buttons. Our application uses computer vision techniques to automatically detect grocery items, helping the user quickly locate the items of interest. In addition, our application uses natural language processing techniques to categorize the sentences to help the user quickly find the desired sentence. We conducted case studies to evaluate our mixed reality-based application with 7 participants with expressive language difficulties to validate the usability and feasibility. To the best of our knowledge, HoloAAC is the very first application that explores context-aware AAC based on mixed reality. We will open-source our holistic implementation.

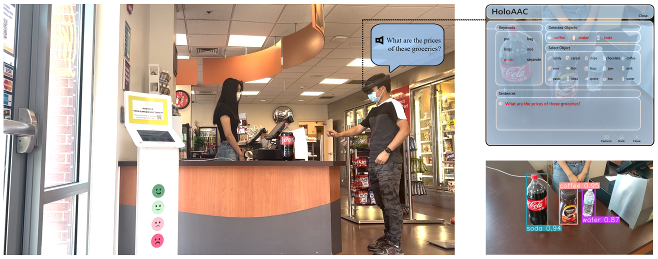


Figure 1: When the user wears HoloLens 2 and stands by the side of the cashier, the user clicks the camera button to capture current objects on the desk. In this scenario, there are three objects on the desk: soda, coffee, and water. After the captured image is processed on the server, the detected objects, the generated keywords, and the generated sentences will be shown in front of the user via an AAC interface visualized by HoloLens 2. As the user clicks the *prices* keyword, the sentence "what are the prices of these groceries?" is shown. The user clicks this sentence to trigger our application to speak it accordingly.

1 Introduction

Augmentative and alternative communication (AAC) Beukelman et al. (1998) is a communication mechanism for those with complex communication needs (CCN) Porter et al. (1995), and existing AAC devices are forms of assistive technology comprising hardware and software that can support or replace natural speech entirely. On the other hand, augmented reality (AR), a user's visual perception supplemented with additional computer-generated sensory modalities, is rising in its abilities to support assistive technology through rehabilitation therapies that support people with visual impairments.

While immersive learning applications in augmented reality have greatly supported individuals with disabilities, current AAC devices do not carry the contextual intelligence to prompt appropriate conversation choices or phrases based on a user's environment. This is particularly concerning for emergency situations where real-time communication is important for supporting AAC users who have to not only consider their accommodations but also navigate a crisis with heightened emotions. This prompts the need for an AI-driven AAC system aware of the environmental situation and demand. Because of the monumental shift of the nature of AAC, AAC has expanded its reach to include more people with a wider range of CCN Shane et al. (2012).

Augmented reality is becoming popular in various fields such as teaching Tzima et al. (2019), learning Mystakidis et al. (2022), entertainment Hung et al. (2021), defense Wang et al. (2020), and marketing Rauschnabel et al. (2019). As an immersive technology, AR opts to observe the user's surroundings, understand the context, and synthesize context-aware content with the aid of computer vision and artificial intelligence algorithms. For instance, running on an AR headset, our HoloAAC app automatically recognizes products in the user's surroundings while the user navigates a grocery store, retrieving relevant conversation sentences. Moreover, head-mounted AR headsets feature egocentric vision, referring to being capable of seeing what the user sees. These factors make head-mounted AR headsets promising vehicles for delivering AAC applications in the future. Compared to current AAC devices that require users to operate an AAC application on a phone or tablet, an AAC app running on a head-mounted AR headset could be less distracting and more intuitive to provide in-situ conversation help.

To explore this direction, we propose a computer vision-guided mixed-reality AAC application that helps AAC users in grocery shopping scenarios as shown in Figure 1. First, we devise a computational approach to generate shopping-related sentences. Second, we use a mixed reality device to capture an image of the current context, based on which an object detection algorithm is applied. Third, we propose a natural language processing (NLP) based algorithm to help the user quickly find the desired sentence. Fourth, a text-to-speech engine will translate the entire sentence into speech upon the user's selection. To the best of our knowledge, HoloAAC is the very first application that explores using mixed reality and contextual-awareness to provide AAC for users with expressive language difficulties. The major contributions of this work include:

- Proposing a novel augmentative and alternative communication interface that can be used on a mixed reality headset;
- Devising an interactive approach based on object detection and text retrieval techniques to help AAC users quickly retrieve and speak desired sentences via text-to-speech;
- Evaluating our approach through experiments that mimic grocery scenarios and case studies conducted with people who have expressive language difficulties.

2 Related Work

There are needs for just-in-time communication and context-aware technologies in the AAC community. In fact, this is an area of needs that has been prevalent. We review some existing works.

2.1 Context-Aware AAC

Communication depends on context. People talk about things that are rooted in their environments Panchanathan et al. (2018). A context-aware system decides what information and which service should be presented to the user Sezer et al. (2018).

TalkAbout Kane et al. (2012) is a context-aware, adaptive AAC system that provides its users with a word list adapted to their current location and conversation partner. TryTalk Ghatkamble et al. (2014) operates similarly, considering the user's location obtained through GPS or building QR code, as well as the day and time. Chan et al. Chan et al. (2016) used the Bluetooth Low Energy beacons to achieve accurate indoor tracking and a micro-location context-aware AAC system to reduce the cognitive load of user interaction. Chan et al. Chan et al. (2020) proposed a context-aware AAC system to facilitate daily communication for nonverbal school children with moderate intellectual disabilities. Vargas de Vargas (2020) proposed a preliminary idea about design and evaluation of a context-adaptive AAC application, which uses SNOW De Deyne et al. (2019) data, ConceptNET Speer et al. (2017), DeScript Wanzare et al. (2016), user-specific semantic network, and clustering algorithms to infer the suggested vocabulary to enable the user to easily find appropriate words when communicating. Shen et al. Shen et al. (2022) devised KWickChat for nonspeaking individuals with motor disabilities, which leverages a GPT-2 language model and context information to improve the quality of the generated responses. Rocha et al. Rocha et al. (2022) proposed a system for people with aphasia, which supports two-way communication. They demonstrated how it can be used by a person with aphasia lying in bed to communicate with a caregiver via a smartwatch.

Unlike the previous works, our application offers full sentences for users to select instead of a single word or a phrase. Inspired by TryTalk Ghatkamble et al. (2014), our application also prioritizes frequently clicked sentences relevant to the detected objects. Our application leverages the image capturing, hand tracking, visualization, and audio capabilities of the HoloLens 2 to realize a novel and integrated AAC interface in augmented reality.

2.2 Computer Vision-based AAC

Computer vision has been applied for AAC. The computer vision-based AAC applications primarily lie on eye tracking, blink recognition, head tracking, facial detection, and sign language recognition Panchanathan et al. (2018).

Eye-Tracking. We review some systems that use eye-tracking to support AAC. Raudonis et al. Raudonis et al. (2009) proposed relatively inexpensive eye-tracking system, which used a web camera and principal component analysis and an artificial neural classifier to achieve the goal of eye-tracking. Al-Rahayfeh et al. Al-Rahayfeh and Faezipour (2013) presented a survey about eye-tracking and head movement, which shows that the eye-tracking can applied in assistive technologies to improve accuracy and lower cost.

Jen et al. Jen et al. (2016) proposed a wearable eye-gaze tracking system that only required one single webcam mounted on the glasses, whose experiments show its high accuracy and robustness. On the other hand, Al-Kassim et al. Al-Kassim and Memon (2017) designed a scanning keyboard to help people with paralysis, which relies on detection and tracking of the user's eyeball movements. Moreover, Zhang et al. Zhang et al. (2017) developed an eye gesture communication system GazeSpeak that can run on a smartphone to help people who have motor impairments. Fiannaca et al. Fiannaca et al. (2017) presented AACrobat, a Gaze-Based AAC to lower communication barriers and provide autonomy using mobile devices. For more recent and detailed research on eye-tracking, please refer to a recent review Klaib et al. (2021).

Sign Language Recognition. Another area of research uses sign language recognition to drive AAC applications. Sign language, e.g., American Sign Language (ASL), is an ideal way to communicate for people who are deaf or hard of hearing Panchanathan et al. (2018).

Akmeliawati et al. Akmeliawati et al. (2007) proposed an automatic vision-based sign language translation system to translate Malaysian into English in real-time. Dreuw et al. Dreuw et al. (2012) presented another sign language recognition and translation system, which is based on statistical machine translation, speech recognition, and image processing, supporting the recognition of complete sentences in sign languages. Halim et al. Halim and Abbas (2015) developed a system for detecting and understanding sign language gestures to assist people with hearing and speech impairments, which employed the dynamic time warping algorithm and Microsoft Kinect. Besides the above computer vision-based sign language recognition research, there are some researches using accelerometers, gyroscopes, and surface electromyography sensors Li et al. (2010); Su et al. (2016); Wei et al. (2016).

Disparate previous AAC research that used computer vision for communication purposes, we leverage computer vision to drive our application: object detection analyzes the context in a scenario, and the detection result hints what items the user is probably concerned about, helping the user quickly generate context-aware sentences.

2.3 Computer Vision for Non-AAC Users

Computer vision can be used for assistive healthcare, which is not limited to AAC users. For example, computer vision can assist visually impaired people to navigate in indoor space, aid people with cognitive impairments, help with neurorehabilitation of post-stroke patients, support the surgeon, and push the development of social robots which are designed to foster people's cognitive and socio-emotional well-being Marco and Farinella (2018).

2.4 Augmented Reality for AAC

Augmented reality for AAC is a relatively new research field. Ramires et al. Ramires Fernandes et al. (2014) discussed an augmented reality based interacting system, which integrated AAC and applied behavior analysis (ABA), to support interventions with children who have Autism Spectrum Disorders (ASD). Also, other researches Bai et al. (2015); Chen et al. (2015, 2016); Cihak et al. (2016); Mcmahon et al. (2015); Liu et al. (2017); Taryadi and Kurniawan (2018) show that AR can be used to improve language and communication skills in individuals with ASD and has positive outcomes such as increased motivation, attention, and learning new tasks Hayden et al. (2017).

Recently, Zheng et al. Zheng et al. (2017) proposed a communication enhancement system KinToon which uses a projector to project cartoon masks to a human face to enable autistic children to interact with their favorite cartoon characters face to face. On the other hand, Kerdvibulvech et al. Kerdvibulvech and Wang (2016) proposed a three-dimensional augmented reality based human-computer interaction application to assist children with special problems in communication.

The direction of using a HoloLens for AAC applications is relatively unexplored. Zhao et al. Zhao et al. (2021) proposed an AAC application that runs on HoloLens to used eye-gaze technology to select words and make sounds. Another application based on HoloLens was designed to assist people with low vision in wayfinding Zhao et al. (2020).

Compared to previous works, HoloAAC does not act as a supportive tool for therapists. It aims to help AAC users in daily grocery shopping scenarios. Also, it aims to speak a meaningful sentence rather than a word or a phrase.

2.5 User Interface and Interaction for AAC

Several design efforts focused on user interfaces and interactions to support AAC applications. Sobel et al. Sobel et al. (2017) explored the design space of AAC awareness

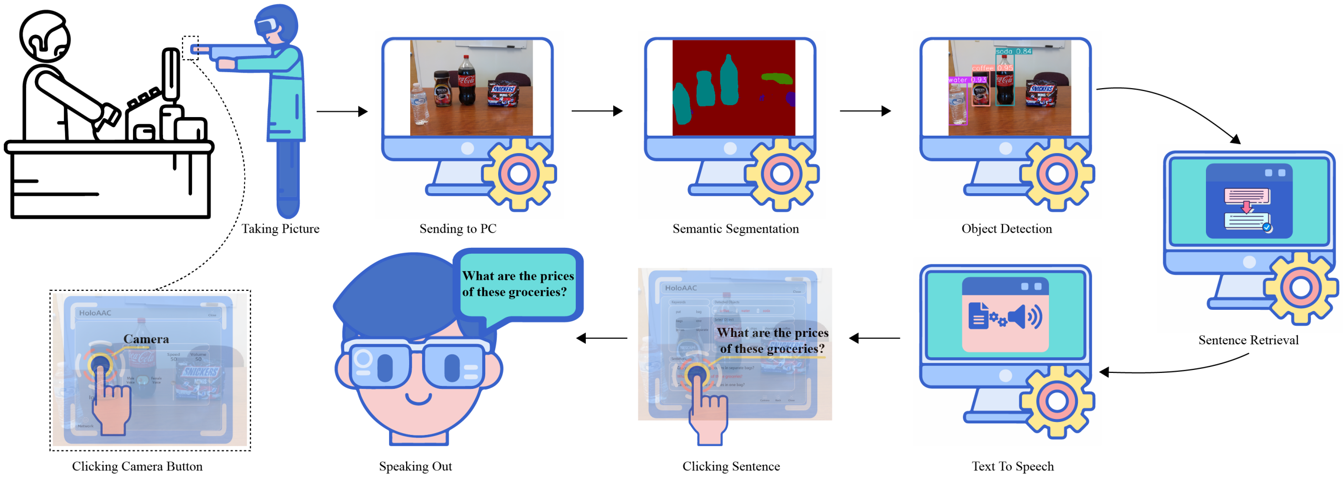


Figure 2: Our application's overview. As a user wearing a HoloLens 2 reaches a cashier, they presses the camera button to capture an image. The image is sent to a server which executes a series of operations: semantic segmentation, object detection, sentence retrieval, and text-to-speech. When HoloLens 2 gets the response from the server, the user can select a desired sentence, triggering our application to speak the sentence.

displays. Gibson et al. Gibson et al. (2020) extracted design requirements from a clinical AAC tablet application. Kristensson et al. Kristensson et al. (2020) proposed a design engineering approach for quantitatively exploring context-aware sentence retrieval. Besides, Obiorah et al. Obiorah et al. (2021) developed three meal ordering prototype systems for people with aphasia dining in restaurants. Mitchell et al. Mitchell et al. (2022) evaluated a generically optimized keyboard and the ubiquitous QWERTY keyboard among three people with dexterity impairments due to motor disabilities.

In this work, we focus on creating an aided AAC. Aided AACs can be categorized into two groups: low-technology AACs and high-technology AACs. Low-technology AACs do not require a battery or wall plug power supply to operate, while high-technology AACs refer to powered devices, either from a battery or a wall plug power supply Norrie et al. (2021). Some high-technology AACs can run on smartphones and tablets. With the commonality of smartphones and tablets, many AAC applications surge, such as Proloquo2Go, Cboard, TouchChat, QuickTalk, iCommunicate, and SonoFlex. For more details about high-technology AAC, please refer to a recent review Elsahar et al. (2019).

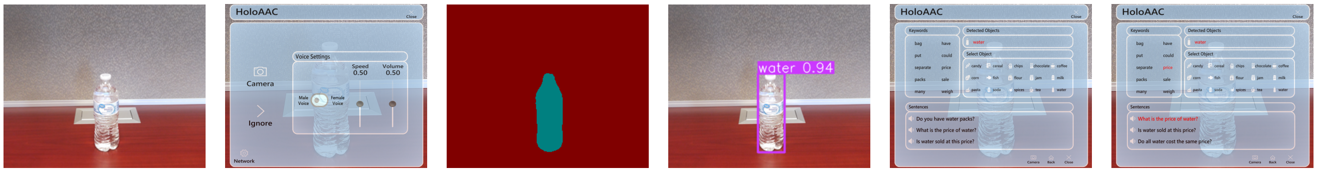
Unlike existing high-tech AAC user interfaces that run on traditional devices, for example, smart phones, tablets, and specially designed electronics, our novel user interface runs on a mixed reality headset, which is portable and wearable, supporting more advanced and immersive interactions.

3 Interview with AAC Users

To devise a friendly, accessible, and practical application for AAC users, we interviewed 2 professional AAC users who have been using AAC devices for more than 3 years and also teaching people to use AAC devices. We obtained the following insights about the design of this application.

- This application should be portable and the device running the application should be untethered.
- This application should be easy to use with minimal configurations and intuitive operations.
- Considering that some AAC users are used to symbol-based or text-based AAC tools, it is preferable to use similar symbols in this application.
- This application should be friendly to those AAC users with listening disabilities.
- For the grocery shopping scenario, it would be convenient to automatically detect items and support the user to select items.

We devise our augmented reality AAC application, HoloAAC, based on the above observations. The application runs on the Microsoft HoloLens 2. It comprises three windows: an entry window, a network setting window, and a main window. In this application, we support setting voice speed, volume, and voice type (male voice/female voice). Besides that, since computer vision can be used in context-aware AAC to determine what objects of interest are in the environment Panchanathan et al. (2018), we use computer vision techniques to detect groceries and provide an optional way to select/deselect groceries. In addition, the application also tracks the user's sentence selection history to prioritize previously selected sentences. Our application employs the wireless network to realize the portable goal. In order to make this application more accessible, we use red color to denote being selected. What's more, we set the pressed



Do you have water packs?
What is the price of water?
Is water sold at this price?

What is the price of water?
Is water sold at this price?
Do all water cost the same price?

(a) (b) (c) (d) (e) (f)

Figure 3: Example with a bottle of water. (a) The scene in front of the user. (b) The HoloLens UI that allows the user to capture an image by pressing the camera button. (c) The semantic segmentation result. (d) The object detection result with the confidence score. (e) After the processing is done on the server end, a user interface is shown via HoloLens 2 to allow the user to select additional keywords and sentences. (f) The window after the keyword "price" was pressed and then the first sentence "what is the price of water?" was pressed. The red colors in the object panel and keyword panel indicate the items being selected, while the red color in the sentences panel refers to the sentence being played.

sentence's color to red to indicate that it is being spoken, which is more friendly for people with listening disability.

4 Overview

Figure 2 shows our application workflow. First, the user wearing a HoloLens 2 takes a picture of the groceries in front. The picture is then sent to the server (a PC in our experiments) for processing: semantic segmentation, object detection, sentence retrieval, and text-to-speech. The user can select one or more keywords to quickly locate the desired sentence and trigger the device to speak it.

Although object detection can be directly employed on the grocery items, detection failure may happen in practice. To enhance the accuracy, we apply a semantic segmentation as a preprocessing step. Semantic segmentation is used to locate the potential object regions in the original pictures taken by HoloLens 2. The potential object regions indicated by bounding boxes will be regarded as the input for object detection. As for the sentence retrieval stage, we add a runtime cache to store the context data, that is, historical clicking data, to help filter sentences. The text-to-speech engine will translate the filtered sentences to audio files, which are sent as the server's response back to the device. After getting the response, the main UI will be updated with the relevant keywords and sentences for the user's further interaction.

Illustrative Example. Taking a water bottle as an example as shown in Figure 3. The user presses the camera button on the UI as shown in Figure 3b. When the capturing is done, it will send a post request to the server, which contains some parameters, for example, male voice/female voice, speech speed, and speech volume. Via a web request, the server gets the captured image. The server will run the semantic segmentation algorithm to identify potential objects regions as shown in Figure 3c. These regions will be processed iteratively. For each iteration, the bounding box of one region will be calculated. The object detection algorithm will be employed to detect an object within the bounding box as shown in Figure 3d. Here "water 0.94" means that the water is detected with a 0.94 confidence score. The confidence score is the product of box confidence score and conditional class probability, which reflects the confidence of localization and classification. The box confidence score refers to the confidence of the box containing an object, while the class probability is conditioned on the bounding box containing an object Redmon et al. (2016).

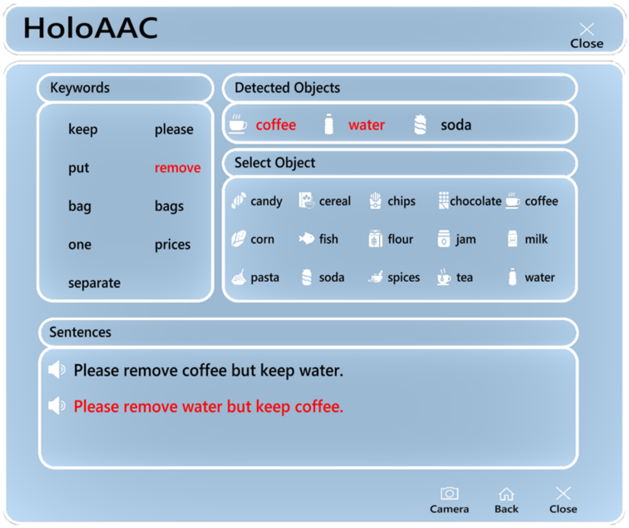
The detected objects' names will be used for the sentence retrieval process. After that, the server will package the data and reply to the HoloLens 2. After receiving the data, the HoloLens 2 will parse the data, and update the UI accordingly as shown in Figure 3e. When the user presses one sentence, the sentence will be spoken by the device. If desired, the user can also press one or more keywords to quickly locate sentences via filtering as shown in Figure 3f.

5 Technical Approach

5.1 AR Tool and User Interface

As aforementioned, our application runs on Microsoft HoloLens 2. We use the Unity and the Mixed Reality Toolkit (MRTK) to develop the application. HoloLens 2 supports hand tracking so the user interface is movable in the 3D space. The user interface primarily includes three parts: Entry UI, Main UI, and Network Setting UI. We discuss the Main UI below and put details of the Entry UI and the Network Setting UI in our supplementary material.

Main UI. The Main UI is where the detected objects, keywords, and sentences show. Figure 4 shows its screenshot. The top *Detected Objects* panel shows the detected objects in the captured picture. The left *Keywords* panel displays keywords related to the selected objects in the *Detected Objects* panel and the *Select Object* panel. The central *Select Object* panel lists all the objects that are



Please remove coffee but keep water.

Please remove water but keep coffee.

Figure 4: Our AAC user interface. Refer to the main text for the functionality description.

supported. In case the object detection fails and therefore no object is detected and automatically selected, the user can still select any object in this panel manually. To enhance understanding, we add a symbol in front of each object's name. The bottom *Sentences* panel shows relevant sentences retrieved according to objects and keywords. When the user presses one sentence, the application will speak the sentence. The speech was generated based on the voices settings, that is, male voice/female voice, speed, and volume as set on the Entry UI.

At the bottom right, there are three buttons: camera, back, and close. This camera button performs the same action as the camera button in the Entry UI. The back button is used to go back to the Entry UI. The close button is used to quit this application. We use a red color to denote the selected objects, keywords, and sentences in the *Detected Objects* panel, the *Select Object* panel, *Keywords* panel, and *Sentences* panel. Note that the *Detected Objects* panel, the *Select Object* panel, and the *Keywords* panel support multi-selection.

5.2 Object Detection

As aforementioned, we take the image captured by the HoloLens 2 as the input. The next step is to detect possible objects on the image.

Object Detection. Inspired by the GroceEye Bhimani (2020), to perform object detection of grocery items, we fine-tune a YOLOv5 model with the Freiburg Grocery dataset. We use the processed Freiburg dataset which can be downloaded from Github.

Semantic Segmentation. As noted, the images of the training dataset in object detection are different from the images captured by the HoloLens 2. The training dataset is cleaner because the image size is small and contains less background information. The images captured by the HoloLens 2 are larger because of the FOV of its camera. As a result, in most cases, directly running the object detection model will yield inaccurate results. Therefore, we propose a preprocessing method to improve object detection precision. In our approach, we first apply a semantic segmentation method (Deeplabv3+) before we perform the object detection.

Object Detection in A Real Grocery Store. As object detection is our approach's entry point, we validate whether it works in a real grocery store scenario. We conducted a preliminary experiment in a real grocery store. To validate the object detection robustness, we run our application from different perspectives as shown in Figure 5. Our proposed object detection approach worked under different lighting conditions (Figure 5a and Figure 5c) and in different angles (Figure 5b and Figure 5c), successfully detecting objects (e.g., soda, water) taken in front, though missing a bag of rice at the side which the user can select manually in an error handling manner (Section 5.4).

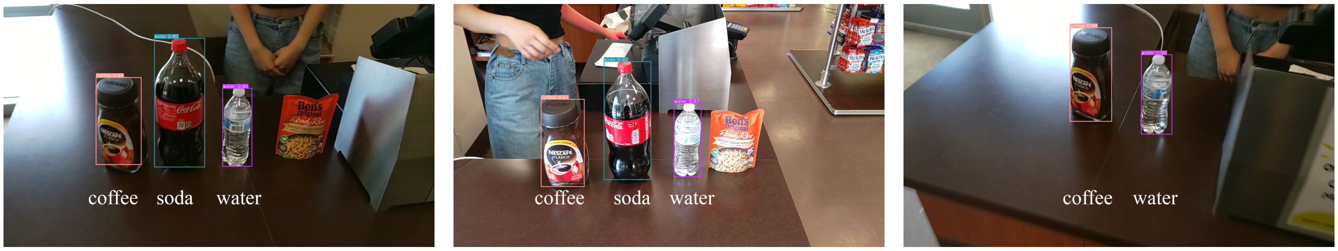
Please refer to our supplementary material for details of fine-tuning, object detection evaluation, semantic segmentation, and the real grocery store object detection experiments.

5.3 Relevant Sentence Retrieval

After detecting the objects on the image, our approach retrieves relevant sentences that the user may want to speak. As illustrated in Figure 6, relevant sentence retrieval can be split into six steps: 1) retrieving object relevant sentences; 2) sentence stemming; 3) keyword generation; 4) sentence grouping; 5) sentence filtering, sorted by historical data; and 6) text-to-speech.

Overall Workflow of Sentence Retrieval. We interviewed two professional AAC users, who have been using AAC devices for more than 3 years and also teaching people to use AAC devices, for their opinion regarding commonly asked questions in grocery shopping scenarios. We abstracted them and made them extensible to support adding other sentences easily. We devise a sentence database to construct object relevant sentences. Since the number of sentences with regards to every object is large, it is hard for a user to locate the target sentence. Therefore, we tokenize and stem sentences to get keywords, which are used to group sentences. Hence the user can select the target sentence through selecting keywords. We also consider historical data, that is, which sentences are selected by the user before, to sort the sentences. As a result, the more times one sentence is selected, the higher the precedence of showing that sentence is. After the sentences are confirmed, the text-to-speech engine will synthesize the corresponding audios of speaking the sentences.

Sentence Dataset Schema Design. We define a database of items to cover most conversation topics in grocery store shopping scenario. Considering the expandability, intuitiveness, and readability, we apply an



(a) Perspective 1's result (b) Perspective 2's result (c) Perspective 3's result

Figure 5: Object detection by our application in a real store. The images were captured by a HoloLens 2 from different perspectives. The labels of the detected objects are shown.

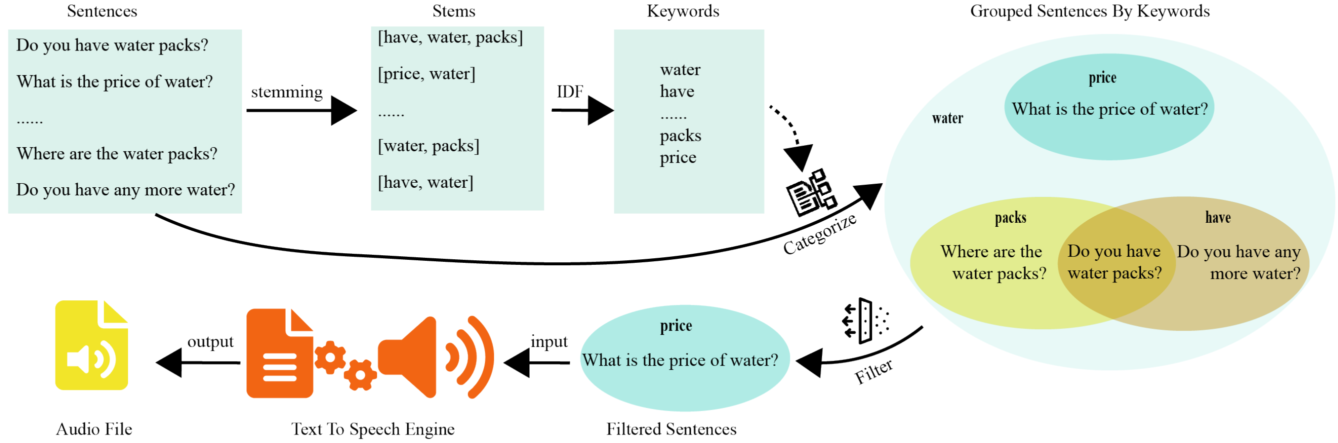


Figure 6: Sentence retrieval overview. Our method first retrieves all sentences containing the detected grocery names. After removing stop words and punctuations, it extracts the stems of each sentence. We use the IDF algorithm to obtain keywords. The sentences will then be categorized by the keywords. The user could click further keywords on the UI, which will then trigger our approach to filter out any irrelevant sentence. Those sentences that pass the filter will be processed by the text-to-speech engine to generate the audio files of the spoken sentences.

object-oriented programming methodology to design this database. It is easy to imagine that most items share some common topics. For example, customer may want to ask about the price of some items, where to find what, and whether something is on sale. Suppose that one customer asks about the prices of cake or water. The user might say "what is the price of cake?" or "what is the price of water?". In this case, the difference between these two sentences is that the subject varies. Therefore, we abstract these sentences to a simple format, that is, what is the price of {name}, where {name} refers to one specific item's name here.

Think about another scenario. The customer may buy several items, for example, cakes, and want to put these items in one bag. In this situation, the user may say "put all the cakes in one bag.". Since cake is a countable noun, it should use its plural form for grammar correctness. Hence, we represent the plural nouns with {names} rather than {name} for countable nouns. Besides that, some items, such as milk, may be more complicated. There are many kinds of milk, like skim milk, non-fat milk, and whole milk. In this case, we define that these items can be accompanied by adjectives. Each of the adjectives can be applied before the item's name.

Suppose the user selects two kinds of milk, for example, skim milk and whole milk, and wants to ask a question about the price of the whole milk. Without the adjective, the abstracted sentence should be "what is the price of this whole {name}?". If they also wants to ask about the price of the skim milk, the abstracted sentence should be "what is the price of this skim {name}?". It is obvious that this design is redundant, so we abstract this case as "what is the price of this {name}?". As milk can carry its adjectives, we regard one adjective and one noun as one entity and then render the sentence "what is the price of this {name}?" with this entity.

Figure 7 shows a sentence dataset schema example. *Topics* lists all topics of any object. For example, *Price* is a topic. It has two *parameters*: *name* and *names*. *Name* denotes the singular format of one object, while *names* denotes its plural format. *Sentences* lists all sentences of this topic. *Name* in sentences will be replaced by one object's singular format. *What* denotes what this file is about. *Topics* under *milk* shows what topics milk owns. *Adjectives* depicts what adjectives can be used for this object. The adjective will be placed in front of the name. In this case, the *name* will be replaced by *nonfat milk* once and also by *1% milk* once. *Sentences* under *milk* indicates those sentences that are only owned by milk. All sentences in the topics included in the object file and all sentences that are within the object file will be translated as the full set of sentences of this object. Our supplementary material conains implementation details.

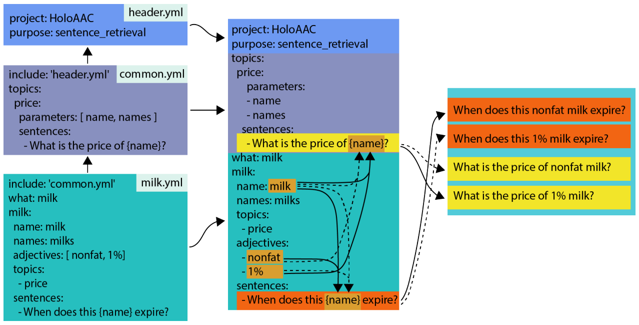
Sentence Dataset Generation. We use the YAML to design a data structure to encode the item-relevant sentences, but the data structure can not be employed directly. We need to design an engine to manipulate these data. Since these files exhibit a parent-child inheritance relationship, in other words, a hierarchical relationship, we apply a Depth First Search algorithm to parse these files. Since we take object names to retrieve relevant sentences, we build a dictionary to store items (keys) and sentences (values).

Besides the single item, we also consider two other scenarios: one is about dual items, the other is about many (more than two) items. We refer these cases as non-single items. For the dual items' scenario, following the same schema defined before, we use {one} and {two} as the parameters. Therefore, the sentences will contain two placeholders, {one} and {two}. One sample sentence is "can you put the {one} and {two} in one bag?". For the many items' scenario, we do not define any parameters. Instead, we use *these groceries* to refer them as a whole. One sample sentence is "can you put these groceries in one bag?".

We apply a preprocessing step to speed up the response during real requests. For each item, we calculate a number of keywords with the IDF algorithm. These keywords can be used to filter the sentences. One keyword forms a group in which all sentences have the keyword. For the many items' scenario, we calculate its keywords. However, for the dual items' scenario, we cannot get the keywords before the request because the sentences are neither set nor deducible.

Keywords Generation for Locating Sentences. As we already have the sentences of one or more items, the next step is to enable the user to quickly select the target sentence. First, for every sentence, we tokenize the sentence, removing punctuations and stopwords. In NLP, stopwords refer to those words that do not add much meaning to a sentence, such as "a" and "the". After that, we get the stem for every sentence. Then, we vectorize the sentences based on the occurrences of words. The result will be a count matrix. We apply the IDF algorithm to get the words with high frequency. In NLP, IDF means inverse document frequency. IDF is a common term weighting schema in information retrieval. A token with a higher IDF weight has a lower frequency, and vice versa. In our approach, we use the top-ten lower IDF weight tokens as the keywords. It will split sentences into several groups.

Sentence Filtering. After we get both the object name(s) and the keywords, we are able to filter the sentence database. First, we filter the subset of the entire sentence dataset using the object name(s). Sentences irrelevant with the objects will be removed, while those relevant will be kept. Then, we filter the subset again with the keywords. After that, we obtain several target sentences that the user may prefer. In order to adapt to the user and personalize our approach, we record the sentences the user has selected before. This data is a kind of prior knowledge. When the user selects the same objects and the same keywords next time, the sentences will be sorted according to this data. The more times a sentence has been selected, the higher precedence of appearance the sentence is given.



```
project: HoloAAC
purpose: sentence_retrieval
```

```
include: 'header.yml'
topics:
  price:
    parameters: [ name, names ]
    sentences:
      - What is the price of {name}?
```

```
include: 'common.yml'
what: milk
milk:
  name: milk
  names: milks
  adjectives: [ nonfat, 1%]
  topics:
    - price
  sentences:
    - When does this {name} expire?
```

```
project: HoloAAC
purpose: sentence_retrieval
topics:
 price:
    parameters:
    - name
    - names
   sentences:
    - What is the price of {name}?
what: milk
milk:
 name: milk
 names: milks
 topics:
  - price
 adjectives:
  - nonfat
  - 1%
 sentences:
  - When does this {name} expire?
```

When does this nonfat milk expire?
When does this 1% milk expire?
What is the price of nonfat milk?
What is the price of 1% milk?

Figure 7: Sentence dataset schema example. We use *include* to indicate the predecessor-successor relation between two files. The including file (the successor) will own all attributes of the included file (the predecessor).

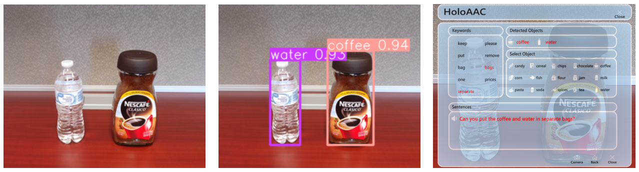
Suppose there are three sentences (S1, S2, and S3) listed on the sentences panel for a specific input when our application runs for the first time. All sentences carry the same frequency 0. Suppose the sentences are shown in the order S1, S2, and S3, and the user selects S2. If our application gets the same input (with the same selected objects and selected keywords) next time, the sentences will be shown in the order S2, S1, and S3.

Text-to-Speech. There are many off-the-shelf solutions for performing text-to-speech. Our approach uses a lightweight offline solution named pyttsx3[1]. It supports male and female voices, changing the speaking speed, and changing the speech volume. This solution outputs audio files in wav format. In order to reduce the audio file size, we convert the format from wav to ogg. In our experiments, for the same data and same sample rate, the file size in wav file is about 79KB, while the corresponding ogg file is just 9KB. It costs less time to transfer the smaller files from the server to the headset.

5.4 Error Handling

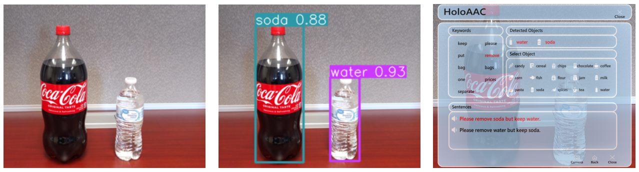
Computer vision techniques such as semantic segmentation could fail in some circumstances, for example, due to motion blur caused by the user's head movement or varying light conditions. We devise our application to tolerate such situations if semantic segmentation or object detection fails. In such situations, our approch leaves the *Detected Objects* panel empty and fill the sentences

[1] https://pypi.org/project/pyttsx3/



(a) Input (b) Segmentation (c) Main UI

Figure 8: Putting two items in separate bags. (a) The image captured. (b) The object detection result. (c) The main UI showing that the *coffee* and *water* were detected, and the *separate* and *bags* keywords were selected by the user. The target sentence *Can you put the water and the coffee in separate bags?* was selected and spoken.



(a) Input (b) Object detection (c) Main UI

Figure 9: Removing one item from two items. (a) The image captured. (b) The object detection result. (c) The main UI showing that the *water* and *soda* were detected, and the *remove* keyword was selected by the user. The target sentence *Please remove soda but keep water.* was selected by the user and spoken.

panel with sentences with no object specification. The user can select listed objects in the *Select Object* panel to retrieve relevant sentences.

6 Experiments and Results

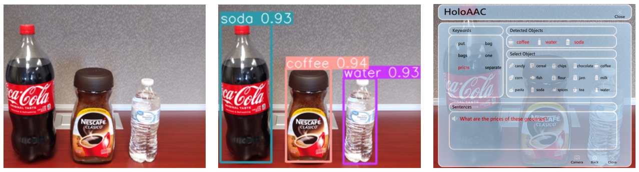
6.1 Implementation

We developed a prototype with a PC installed with Unity, Microsoft Visual Studio 2019, Anaconda3, and PyCharm 2021.2.3. The web service also runs on this PC. The prototype runs on a Microsoft HoloLens 2. For fine-tuning the YOLOv5 object detection model, we used a PC with a Nvidia GTX 3090 graphics card.

6.2 Different Scenarios

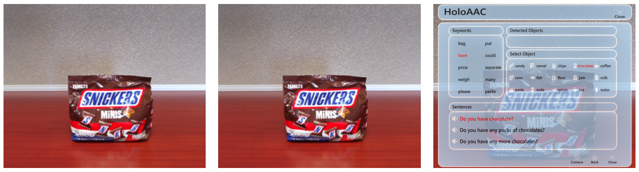
We created four scenarios to simulate potential sentences to say at a grocery store cashier, which are described as follows.

Putting Two Items in Separate Bags. Figure 8 shows this scenario. One bottle of water and one bottle of coffee are on the desk. The objective is to say "can you put the coffee and the water in separate bags?". Based on the captured image, our application managed to detect the water and coffee, and retrieved the relevant sentence. The user clicked on this sentence to trigger the device to say it.



(a) Input (b) Object detection (c) Main UI

Figure 10: Asking about item prices. (a) The image captured. (b) The object detection result. (c) The main UI showing that the *coffee*, *water*, and *soda* were detected, and the *prices* keyword was selected by the user. The target sentence *What are the prices of these groceries?* was selected and spoken.



(a) Input (b) Object detection (c) Main UI

Figure 11: Object detection failure. (a) The image captured. (b) The object detection result. It did not succeed as nothing was segmented in the previous step. (c) The main UI showing that the *chocolate* and *have* keywords were selected by the user. The target sentence *Do you have chocolate?* was selected and spoken.

Removing One Item From Two Items. Figure 9 shows this scenario. One bottle of soda and one bottle of water are on the desk. The goal is to ask the cashier to remove the soda but keep the water. Based on the captured image, the application detected both items. Two relevant sentences were retrieved. The first sentence "please remove soda but keep water." was selected by the user and spoken by the device.

Asking about Item Prices. Figure 10 shows this scenario. One bottle of soda, one bottle of coffee, and one bottle of water are on the desk. The objective is to ask about the item prices. Based on the captured image, our application detected all the objects in the window. The sentence "what are the prices of these groceries?" was retrieved, selected by the user, and spoken by the application.

Object Detection Failure. Figure 11 shows this scenario. One bag of chocolate is on the desk. The objective is to ask if chocolate is available. Our application failed to detect this object based on the captured image. However, the user manually selected the keywords "chocolate" and "have" in the panels. Three relevant sentences were retrieved, of which the first sentence "do you have chocolate?" was selected by the user and spoken by the application.

7 Case Studies

As disability simulations might introduce negative stereotypes and fail to highlight infrastructural and social challenges Bennett and Rosner (2019), we recruited people with expressive language difficulties for case studies. According to the American Speech-Language-Hearing Association, about 0.60% of the population use AAC [2]. Inspired by AACrobat Fiannaca et al. (2017), we formed case studies where we observed a small group of people with expressive language difficulties who used HoloAAC to complete tasks. We then obtained the users' feedback. According to the local standards for sample size in computer-human interaction studies Caine (2016), considering the COVID-19 pandemic, the study setting, and the availability of participants, we recruited 7 participants. This sample size follows the highly expert recommendations ranging from 4 ± 1 to 10 ± 2 Caine (2016). P1, P2, P3, and P4 are local and came to our lab for their case studies. P5 lives in another state, which is about 400 miles away from our lab. P6 and P7 are also non-local and they come from an aphasia rehabilitation center in another state, which is about 100 miles away from our lab.

Since Proloquo2Go[3] (Figure 12) is a popular AAC application on iPhone and iPad for people with expressive language difficulties DongGyu et al. (2014), we let participants complete the same tasks using it as a baseline to investigate the usability and feasibility of our application. Considering the comfort, IRB regulation, safety, convenience, and privacy of AAC users, we conducted the case studies in a simulated environment for P1, P2, P3, and P4. We used a private room inside a lab and set up an environment similar to a grocery store cashier. As for P5, we drove to his home to conduct the case study. Similarly, we drove to the rehabilitation center to conduct the case studies for P6 and P7.

7.1 Participants

We recruited 7 participants for the case studies, five (P1, P2, P3, P4, and P5) of whom are AAC users with 5˜20 years AAC experience, and two (P6 and P7) of whom have aphasia and use phone typing to help themselves with speaking. P1, P2, P3, P4, and P5 used AAC devices mostly at home, in school or work, and in the community. P1, P2, P3, and P4 are local, and they came to the lab for the case studies. P5, P6, and P7 are non-local. P5 lives in another state, and P6 and P7 are from a rehabilitation center in another state. Participant ages ranged from 18 to 74 years old. Education demographics show participants who received less than a high school diploma to a doctorate-level degree. Three participants were working full-time, two were unable to work, one was looking for work, and one was retired.

P1 (Female, Proloquo4Text, 5 years of AAC experience). P1 is an AAC user. She is blind in her right eye. She is proficient in using iPhone and iPad to communicate. She has used Proloquo4Text[4] for about 5 years. She types on a phone or a tablet to communicate in daily life. She has good physical coordination and motion control ability with her fingers. She does not have any VR/AR experience.

P2 (Male, ASL Interpreter, 5 years of AAC experience). P2 is an AAC user. He is deaf. He used to use a teletypewriter to communicate with people online. Then he started to use a webcam which allows him to communicate face to face with others in ASL. In his daily life, he uses an ASL interpreter or type to communicate with others. Unlike P1, He has prior experience with VR headsets and AR apps.

P3 (Female, EZKeys, 20 years of AAC experience). P3 is an AAC user. She is a non-native English speaker. She not only types on phones and computers but also uses an AAC tool called EZKeys[5] for 20 years. She does not have any experience with VR/AR.

P4 (Male, Proloquo2Go, 11 years of AAC experience). P4 is an AAC user. Due to a lower-limb disability, he uses a mobile scooter for daily transportation. He has been using Proloquo2Go iPad and iPhone Apps for 11 years. He has prior VR experience but no AR experience.

P5 (Male, NovaChat 8[6], 5 years of AAC experience). P5 is an AAC user. He lives with his family and uses a symbol-based AAC device NovaChat 8 for daily communication (e.g., home and shopping). He did not use Proloquo2Go before the case study. He does not have prior VR/AR experience.

P6 (Male, Cellphone/iPad, 0 years of AAC experiences). P6 has aphasia. He usually communicates with others through his cellphone or iPad. He did not use Proloquo2Go before the case study. He has some experience in VR games.

P7 (Male, Cellphone, 0 years of AAC experiences). P7 has aphasia and hemiplegia. He is not able to move his right hand because of hemiplegia. He is used to typing on his cellphone to express his thoughts. He neither used Proloquo2Go nor AR/VR devices.

7.2 Procedure

Control Groups. We used two control groups: Proloquo2Go Symbol (Figure 12a) and Proloquo2Go Typing (Figure 12b) since these two modes are frequently used by AAC users. In our case study, Proloquo2Go runs on an iPad.

Warm-up Session. We conducted a warm-up session to get participants familiarized with the basic operations of Proloquo2Go and our application as well. To let them get ready for the formal case study tasks, the warm-up

[2] https://www.asha.org/njc/aac/

[3] https://www.assistiveware.com/products/proloquo2go

[4] https://www.assistiveware.com/products/proloquo4text

[5] http://www.techaccess4you.com/ez-keys-software/

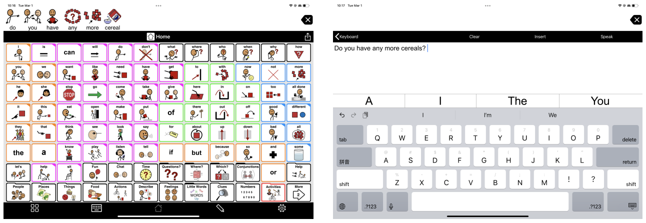
[6] https://store.prc-saltillo.com/novachat-8-active-f19015-30

Table 1: Target sentences used for the six tasks. To avoid confusion, we used *bag* in Proloquo2Go Typing and HoloAAC, and *plastic bag* in Proloquo2Go Symbol as the *bag* symbol in Proloquo2Go was not a plastic bag. Also, as Proloquo2Go did not have the plural form symbol of *bag* and *soda*, we used the singular form. Besides, in Proloquo2Go Symbol, we omitted the punctuations of the target sentences for simplicity.

Task	**Item(s)**	**Proloquo2Go Typing and HoloAAC**	**Proloquo2Go Symbol**
1	water	What is the price of water?	What is the price of water
2	soda	Do you have six-packs of soda?	Do you have six-packs of soda
3	coffee	Do you have any more coffee?	Do you have any more coffee
4	soda	Put all the sodas in one bag.	Put all the soda in one plastic bag
5	water, soda	Can you put the water and soda in one bag?	Can you put the water and soda in one plastic bag
6	water, coffee, soda	Can you put these groceries in separate bags?	Can you put these groceries in separate plastic bag

Table 2: Task completion times (Unit: second) of the participants. HL, PS, and PT denote the HoloAAC, Proloquo2Go Symbol, and Proloquo2Go Typing conditions.

Participant	**Task 1**			**Task 2**			**Task 3**			**Task 4**			**Task 5**			**Task 6**		
	HL	**PS**	**PT**	**HL**	**PS**	**PT**	**HL**	**PS**	**PT**	**HL**	**PS**	**PT**	**HL**	**PS**	**PT**	**HL**	**PS**	**PT**
P1	**18**	63	20	**12**	32	19	30	47	**16**	21	90	**15**	**11**	84	20	**18**	81	23
P2	40	66	**9**	12	65	**8**	40	24	**9**	32	86	**14**	52	102	**18**	27	93	**23**
P3	**33**	84	34	**15**	137	22	78	48	**23**	39	147	**23**	**15**	147	30	**39**	114	43
P4	10	92	**7**	35	47	**21**	43	41	**5**	60	113	**7**	14	107	**9**	10	91	**16**
P5	27	134	**23**	**15**	116	32	33	143	**27**	33	214	**24**	**14**	190	38	**19**	211	51
P6	**39**	153	81	**26**	75	82	**38**	214	64	**27**	245	47	**7**	256	64	**15**	105	123
P7	**12**	156	16	31	206	**29**	35	100	**22**	**30**	221	50	41	84	**38**	**17**	119	57



do you have any more cereal

Do you have any more cereals?

(a) Proloquo2Go Symbol (PS) (b) Proloquo2Go Typing (PT)

Figure 12: Screenshots of Proloquo2Go Symbol and Typing.

session comprised of two tasks. The two warm-up tasks were the same, except that we assisted them to finish the first task while they finished the second task independently. For counterbalancing, the participant did the tasks in different orders. For example, if the participant did Proloquo2Go Symbol, Proloquo2Go Typing, and HoloAAC for the first task, the participant would do the second warm-up task in a different order: e.g., HoloAAC, Proloquo2Go Symbol, and Proloquo2Go Typing.

Case Study Tasks. As shown in Table 1, we designed 6 tasks with different target sentences, which were also given with counterbalancing. Our application tracked the time spent on different operations (e.g., clicking keywords). However, as Proloquo2Go does not have a timing function, we employed an external timer to count the time for the Proloquo2Go Symbol and Proloquo2Go Typing conditions. For Proloquo2Go Typing, we ended the timer once the user has typed the entire sentence. For Proloquo2Go Symbol, we ended the timer once the user has typed the last symbol.

Questionnaire. After the last Proloquo2Go Symbol/Typing and HoloAAC tasks were done, we asked the participant to finish a questionnaire to evaluate the workload. We used the NASA Task Load Index (TLX) Hart (1986) to get the subjective workload assessment. It has six questions in total, which are answered using a 7-Likert scale.

Table 3: Task completion times analysis. HL, PS, and PT denote the HoloAAC, Proloquo2Go Symbol, and Proloquo2Go Typing conditions. SD denotes standard deviation.

Participant	**Metrics**	**HL**	**PS**	**PT**
P1	Mean	**18.33**	66.08	18.82
	SD	6.89	23.15	**3.06**
P2	Mean	33.83	72.78	**13.45**
	SD	13.66	27.98	**5.98**
P3	Mean	36.50	113.08	**29.21**
	SD	23.11	39.85	**8.40**
P4	Mean	28.67	81.85	**10.90**
	SD	20.68	30.64	**6.47**
P5	Mean	**23.50**	167.98	32.58
	SD	**8.67**	42.07	10.81
P6	Mean	**25.33**	174.60	76.80
	SD	**12.60**	75.20	25.93
P7	Mean	**27.67**	147.58	35.45
	SD	**11.02**	56.47	15.93

7.3 Result Analysis

Table 2 shows the task completion times of the participants. Table 3 shows the mean and standard deviation of the completion times. We can see that P1, P2, P3, and P4 show a more stable ability to type, probably because

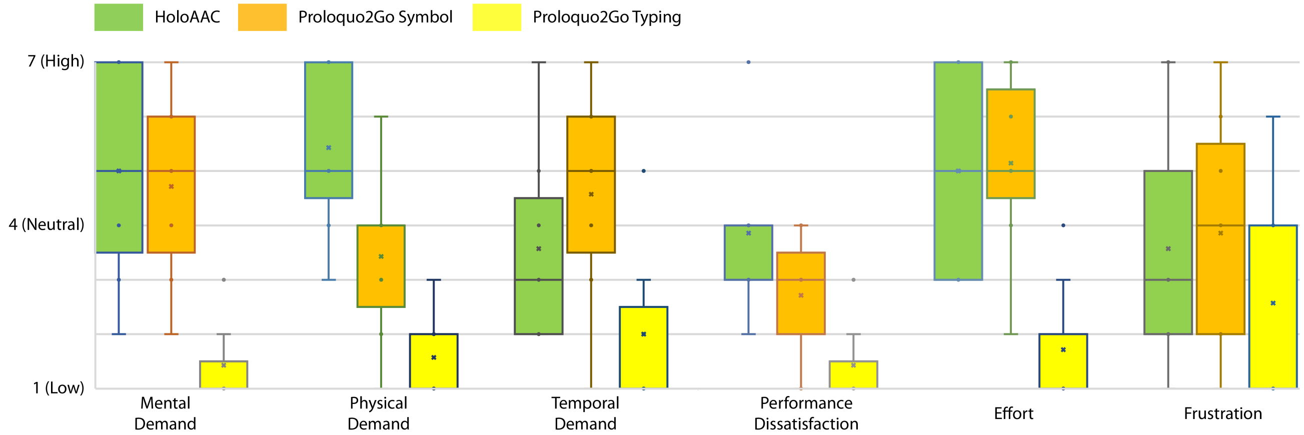


Figure 13: NASA TLX workload assessment rating plots. Each box and whisker plot comprises six-number summary of the rating: minimum, lower quartile (Q1), median (line), mean (×), upper quartile (Q3), and maximum. Please refer to Section 7.4 for the findings and explanations.

Table 4: Mean completion time for each task with HoloAAC.

Task	1	2	3	4	5	6
Mean Time(s)	26	21	42	35	22	21

they type frequently in their daily life. During the case study, they sometimes chose the autocomplete words supplied by the tablet's input keyboard to speed up their input; so sometimes they did not type the whole word.

P1 took similar time using HoloAAC or Proloquo2Go Typing. The completion times with HoloAAC are less than those with Proloquo2Go Typing in 4 out of 6 tasks (Task 1, 2, 5 & 6).

P2 took more time using HoloAAC than Proloquo2Go Typing. We found that it was hard for him to quickly manage to click the target sentence in AR. It took him many attempts to click one sentence to make it speak.

P3 took slightly more time using HoloAAC than Proloquo2Go Typing on average. However, she finished 4 out of 6 tasks (Task 1, 2, 5 & 6) faster using HoloAAC.

P4 took more time using HoloAAC than Proloquo2Go Typing probably due to his many years of experience with Proloquo2Go but no experience with AR.

P5 took less time in 3 out of 6 tasks (Task 2, 5 & 6) using HoloAAC than Proloquo2Go Typing. The mean and SD show that using HoloAAC is faster than using Proloquo2Go Symbol or Typing. Proloquo2Go and HoloAAC are both new to him. The data shows that he becomes familiar with HoloAAC faster than with Proloquo2Go.

P6 took less time in all 6 tasks using HoloAAC compared to using Proloquo2Go Symbol or Proloquo2Go Typing. From the SD and mean, using HoloAAC is faster than using Proloquo2Go Symbol or Typing.

P7 took less time in 3 out of 6 tasks (Task 1, 4, & 6) using HoloAAC than Proloquo2Go Typing. From the SD and mean, using HoloAAC is faster than using Proloquo2Go Symbol or Typing. Because of hemiplegia, P7 felt hard in clicking the sentence precisely and gradually became frustrated as the case study went by. As a result, in the NASA TLX, he gave the same ratings for all questions under HoloAAC (7), Proloquo2Go Symbol (4), and Proloquo2Go Typing (1) to finish the case study quickly.

We note that the participants generally finished the tasks much faster using HoloAAC than using Proloquo2Go Symbol, even for P1 and P4 who are experienced with Proloquo2Go but not with AR. It seems that choosing keywords/symbols to finish a sentence exactly may take more time than typing especially for experienced typers.

Table 4 shows the mean completion time for each task. We can see that Task 3 and Task 4 are the top-two in time consumption as they required the participant to click keywords in AR. More AR mid-air interactions generally resulted in more time needed. Our supplementary material contains the detailed breakdowns of the times spent on each operation in each task by each participant.

7.4 User Feedback

General Feedback. About our HoloAAC application, all participants said that they liked the automatic popping up of relevant keywords and sentences with respect to the objects detected.

P1 liked the camera feature which could help her express her thoughts faster. She disliked that she needed more attempts to interact with the AR interface because she was blind in her right eye.

P2 liked the way how sentences can be automatically generated. He disliked that it was sometimes hard to click sentences in AR. His unfamiliarity with AR glasses posed some challenges for him in completing some tasks, but it also made him feel very fulfilled after completing the tasks.

P3 liked the feature of automatically detecting objects and generating sentences. She thought that the response to user input could improve, and she wanted the mid-

Table 5: NASA TLX workload assessment ratings given by the participants. HL, PS, and PT denote the HoloAAC, Proloquo2Go Symbol, and Proloquo2Go Typing conditions. Please refer to Section 7.4 for the findings and explanations.

Participant	Mental Demand			Physical Demand			Temporal Demand			Performance Dissatisfaction			Effort			Frustration		
	HL	PS	PT	HL	PS	PT	HL	PS	PT	HL	PS	PT	HL	PS	PT	HL	PS	PT
P1	4	5	1	3	1	1	4	5	2	3	3	1	3	5	1	1	1	1
P2	5	6	1	5	6	1	2	6	1	4	3	1	5	6	1	2	6	1
P3	7	2	1	7	2	2	3	1	1	3	1	1	7	2	1	2	2	1
P4	2	6	2	4	3	2	2	6	5	2	3	2	3	7	3	3	5	4
P5	7	7	1	7	4	1	2	7	1	4	1	1	7	7	1	7	7	6
P6	3	3	3	5	4	3	5	3	3	4	4	3	3	5	4	3	2	4
P7	7	4	1	7	4	1	7	4	1	7	4	1	7	4	1	7	4	1

air clicking to be smoother. During the case study, she sometimes made multiple attempts to click target sentences. She also felt accomplished after completing all the tasks. She suggested that our application could also be extended for use in hospitals.

P4 liked the speed and efficiency of HoloAAC compared to other AAC products. He made three comments. First, it would be powerful if our approach could be extended to distinguish subtle differences such as colors and sizes between items. Second, it would be helpful to take pictures and generate speeches almost instantly. As an example, if he passed by a cute dog on street, he would want HoloAAC to instantly say “cute dog” with minimal or no selection needed. Third, he thought that HoloAAC could be extended to provide personalized response options by analyzing conversations in a certain context, for example, when he is chatting with a friend, sentences about his recent personal stories could be retrieved.

P5 liked the new interaction approach. He felt excited when he managed to click the expected sentence. He disliked the interaction accuracy because it took him much time and effort to click. He suggested that HoloAAC can be used in school.

P6 was extremely eager to learn and use HoloAAC. He was enthusiastic about this new technology since it can help him communicate with others. On the other hand, HoloAAC can save his time thanks to its ability to recognize objects, because of which he did not have to memorize texts. He hoped that HoloAAC may be enhanced in terms of interaction accuracy. Additionally, he suggested that HoloAAC can be used extensively in parks, shops, and schools.

P7 did not like HoloAAC. He had a hard time trying to click sentences due to his hemiplegia. He was only able to control his left hand to perform clicking actions. He suggested to make the interaction more accurate and sensitive so that it can benefit more people in the workplace.

NASA TLX. We used NASA TLX to measure the workload. It measures the workload from six aspects: mental demand, physical demand, temporal demand, performance dissatisfaction, effort, and frustration. Table 5 shows the original ratings and Figure 13 shows the rating plots using the box and whisker plot. 1 represents very low and 7 represents very high. For each aspect, the lower the rating is, the better.

For P1, P2 and P4, in general, HoloAAC is better than Proloquo2Go Symbol in all six aspects. For P1 and P2, they performed very well with Proloquo2Go Typing probably because they have about 5 years of AAC experience and are proficient at typing (P1 used Proloquo4Text and P2 used a teletypewriter). For P4, HoloAAC is comparable to Proloquo2Go Typing. For P3, she gave high ratings in mental demand, physical demand, and effort for HoloAAC as she experienced difficulties in interacting with the AR interface and she had strong experience (20 years) with traditional AAC devices. For P5, he gave high ratings on mental demand, physical demand, effort, and frustration to HoloAAC. The reason is that because of his myopia, he often adjusted his glasses to try to see the holographics clearly. Besides that, during the case study, we found that he needed many attempts to click the expected sentence. For P6, he gave similar ratings because Proloquo2Go and HoloAAC are both new to him. For P7, he gave the highest ratings to HoloAAC, middle ratings to Proloquo2Go Symbol, and lowest ratings to Proloquo2Go Typing. The reason is that he is only able to use his left hand to complete tasks. Compared to other participants, mid-air AR interactions took him more efforts and led to physical fatigue more easily.

- **Mental Demand.** The average rating of HoloAAC is 5; and 4 out of 7 ratings are greater than 4. The reason is that participants needed to focus on the AR panel to be able to interact. On the other hand, all 7 participants didn’t use HoloLens 2 before, but they were more or less experienced in Proloquo2Go or similar devices/applications. That’s why they gave a high rating to the mental demand.

- **Physical Demand.** The average rating of HoloAAC is 5.43, which is even higher than that of the mental demand. 5 out of 7 ratings are greater than 4. The reason is that the task was simple to understand, but the interaction required motion control. Some of the participants had disabilities besides speaking disabilities, which made the physical demand even higher. Another reason is, as Plasson et al. Plasson et al. (2020) pointed out, mid-air interaction that HoloLens uses is less accurate than 2D touch and tends to result in physical fatigue.

- **Temporal Demand.** The average rating of HoloAAC is 3.57, a little better than neutral (4); and 5 out of 7 ratings are less than or equal to 4. The reason is that participants didn't feel stressful when performing the tasks. On the other hand, few interactions were needed to complete the tasks using HoloAAC.

- **Performance Dissatisfaction.** The average rating of Holo-AAC is 3.86, a little better than neutral (4). 6 out of 7 ratings are less or equal to 4. Note that only P7 gave a high rating (7) for this aspect. The reason is that P7 did attempt many times to interact with the AR interface because of his hemiplegia. We can say most participants tended to be satisfied with their performance.

- **Effort.** The average rating of HoloAAC is 5, which is equal to the mental demand rating. 4 out of 7 ratings are greater than 4. The reason is that participants had other disabilities in eyes or motion control, which required more effort.

- **Frustration.** The average rating of HoloAAC is 3.57, a little better than neutral (4). 5 out of 7 ratings are less than 4. Most participants didn't feel high frustration when performing tasks using HoloAAC.

In all six aspects, participants gave lowest ratings to Proloquo2Go Typing. That is because 26-keys keyboard based typing is common, and the participants were more or less experienced in it.

7.5 Limitations and Future Work

Due to the small AAC population, it was challenging to recruit many participants to evaluate our application. As a result, we are not able to draw statistically meaningful conclusions.

We only demonstrate HoloAAC for simple grocery scenarios. As scene understanding techniques continue to advance, more sophisticated scene and contextual information could be analyzed for driving an AR-based AAC application. For example, the scene background could help determine where the user is situated (e.g., grocery store, bookstore, music store), providing hints for recognizing objects and retrieving sentences relevant to the current scene type. Besides, based on egocentric computer vision techniques, the application could also deduce the current interactions between the user and the surrounding objects or people so as to retrieve relevant sentences to enhance communications.

Another possible extension is to attach a 4G/5G communication module to enable HoloLens to work without Wi-Fi, which would allow our application to be employed in more scenarios such as supporting outdoor activities. Besides, due to the reality that a standard disabled experience rarely plays out in practice Hofmann et al. (2020), it would be helpful to consider multiple disabilities so as to better accommodate AAC users. For example, for those people with both expressive language difficulties and motion control disability, an interaction mechanism based on eye-tracking rather than hand-clicking is more accessible. Moreover, with the emergence of ChatGPT and GPT-4, it would be interesting to integrate them into our approach: after HoloAAC detects objects and the user selects some keywords, ChatGPT/GPT-4 can generate sentences to be selected, which are spoken through a text-to-speech module.

For those users who did not use HoloLens before, it might take them some time to get familiar with the AR interactions. In our case study, some participants experienced difficulty in clicking the keywords or sentences shown in augmented reality. We believe that improving the hand tracking precision would make AR-based AAC applications more practical and favorable. Alternatively, instead of using mid-air interactions, using a controller (e.g., the clicker of HoloLens 1) could make interaction easier especially for users with body movement disabilities.

References

Rini Akmeliawati, Melanie Po-Leen Ooi, and Ye Chow Kuang. 2007. Real-Time Malaysian Sign Language Translation using Colour Segmentation and Neural Network. In *2007 IEEE Instrumentation Measurement Technology Conference IMTC 2007*. 1–6. https://doi.org/10.1109/IMTC.2007.379311

Zeenat Al-Kassim and Qurban Ali Memon. 2017. Designing a low-cost eyeball tracking keyboard for paralyzed people. *Computers & Electrical Engineering* 58 (2017), 20–29.

Amer Al-Rahayfeh and Miad Faezipour. 2013. Eye Tracking and Head Movement Detection: A State-of-Art Survey. *IEEE Journal of Translational Engineering in Health and Medicine* 1 (2013), 2100212 –2100212. https://doi.org/10.1109/JTEHM.2013.2289879

Zhen Bai, Alan Blackwell, and George Coulouris. 2015. Using Augmented Reality to Elicit Pretend Play for Children with Autism. *Visualization and Computer Graphics, IEEE Transactions on* 21 (05 2015), 598–610. https://doi.org/10.1109/TVCG.2014.2385092

Cynthia L Bennett and Daniela K Rosner. 2019. The promise of empathy: Design, disability, and knowing the" other". In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–13.

David R Beukelman, Pat Mirenda, et al. 1998. *Augmentative and alternative communication*. Paul H. Brookes Baltimore.

Rehaan M. Bhimani. 2020. *GrocerEye - A YOLO Model for Grocery Object Detection*. Retrieved April 1, 2021 from http://students.washington.edu/bhimar/highlights/2020-12-18-GrocerEye/

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

Rosanna Yuen-Yan Chan, Xue Bai, Xi Chen, Shuang Jia, and Xiao-hong Xu. 2016. IBeacon and HCI in Special Education: Micro-Location Based Augmentative and Alternative Communication for Children with Intellectual Disabilities. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI EA '16)*. Association for Computing Machinery, New York, NY, USA, 1533–1539. https://doi.org/10.1145/2851581.2892375

Rosanna Yuen-Yan Chan, Eri Sato-Shimokawara, Xue Bai, Motohashi Yukiharu, Sze Wing Kuo, and Anson Chung. 2020. A Context-Aware Augmentative and Alternative Communication System for School Children With Intellectual Disabilities. *IEEE Systems Journal* 14 (2020), 208–219.

Chien-Hsu Chen, I-Jui Lee, and Ling-Yi Lin. 2015. Augmented reality-based self-facial modeling to promote the emotional expression and social skills of adolescents with autism spectrum disorders. *Research in Developmental Disabilities* 36 (2015), 396–403. https://doi.org/10.1016/j.ridd.2014.10.015

Chien-Hsu Chen, I-Jui Lee, and Ling-Yi Lin. 2016. Augmented reality-based video-modeling storybook of nonverbal facial cues for children with autism spectrum disorder to improve their perceptions and judgments of facial expressions and emotions. *Computers in Human Behavior* 55 (2016), 477–485. https://doi.org/10.1016/j.chb.2015.09.033

David F. Cihak, Eric J. Moore, Rachel E. Wright, Don D. McMahon, Melinda M. Gibbons, and Cate Smith. 2016. Evaluating Augmented Reality to Complete a Chain Task for Elementary Students With Autism. *Journal of Special Education Technology* 31, 2 (2016), 99–108. https://doi.org/10.1177/0162643416651724 arXiv:https://doi.org/10.1177/0162643416651724

Simon De Deyne, Danielle J Navarro, Amy Perfors, Marc Brysbaert, and Gert Storms. 2019. Measuring the Associative Structure of English: The 'Small World of Words' Norms for Word Association. *Behavior Resarch Methods* 51, 3 (2019), 987–1006.

Maurício Fontana de Vargas. 2020. Design and evaluation of a context-adaptive AAC application for people with aphasia. *ACM SIGACCESS Accessibility and Computing* (2020), 1 – 1.

Park DongGyu, Sejun Song, and DoHoon Lee. 2014. Smart phone-based context-aware augmentative and alternative communications system. *Journal of Central South University* 21 (09 2014), 3551–3558. https://doi.org/10.1007/s11771-014-2335-3

Philippe Dreuw, Daniel Stein, Thomas Deselaers, David Rybach, Morteza Zahedi, Jan Bungeroth, and Hermann Ney. 2012. Spoken Language Processing Techniques for Sign Language Recognition and Translation. *Technology and Disability* 20 (04 2012). https://doi.org/10.3233/TAD-2008-20207

Yasmin Elsahar, Sijung Hu, Kaddour Bouazza-Marouf, David Kerr, and Annysa Mansor. 2019. Augmentative and alternative communication (AAC) advances: A review of configurations for individuals with a speech disability. *Sensors* 19, 8 (2019), 1911.

Alexander Fiannaca, Ann Paradiso, Mira Shah, and Meredith Ringel Morris. 2017. AACrobat: Using mobile devices to lower communication barriers and provide autonomy with gaze-based AAC. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. 683–695.

Rajlakshmi Ghatkamble, JinHan Son, and DongGyu Park. 2014. A design and implementation of smartphone-based AAC system. *Journal of the Korea Institute of Information and Communication Engineering* 18, 8 (2014), 1895–1903.

Ryan Colin Gibson, Mark D Dunlop, Matt-Mouley Bouamrane, and Revathy Nayar. 2020. Designing clinical AAC tablet applications with adults who have mild intellectual disabilities. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

Zahid Halim and Ghulam Abbas. 2015. A kinect-based sign language hand gesture recognition system for hearing-and speech-impaired: a pilot study of Pakistani sign language. *Assistive Technology* 27, 1 (2015), 34–43.

Sandra G Hart. 1986. NASA task load index (TLX). (1986).

Candace Marie Hayden et al. 2017. *Augmented reality for speech and language intervention in autism spectrum disorder*. Ph.D. Dissertation.

Megan Hofmann, Devva Kasnitz, Jennifer Mankoff, and Cynthia L Bennett. 2020. Living disability theory: Reflections on access, research, and design. In *Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility*. 1–13.

Shiu-Wan Hung, Che-Wei Chang, and Yu-Chen Ma. 2021. A new reality: Exploring continuance intention to use mobile augmented reality for entertainment purposes. *Technology in Society* 67 (2021), 101757.

Cheng-Lung Jen, Yen-Lin Chen, You-Jie Lin, Chao-Hsien Lee, Augustine Tsai, and Meng-Tsan Li. 2016. Vision based wearable eye-gaze tracking system. In *2016 IEEE international conference on consumer electronics (ICCE)*. IEEE, 202–203.

Shaun K. Kane, Barbara Linam-Church, Kyle Althoff, and Denise McCall. 2012. What We Talk about: Designing a Context-Aware Communication Tool for People with Aphasia. In *Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility* (Boulder, Colorado, USA) *(ASSETS '12)*. Association for Computing Machinery, New York, NY, USA, 49–56. https://doi.org/10.1145/2384916.2384926

Chutisant Kerdvibulvech and Chih-Chien Wang. 2016. A New 3D Augmented Reality Application for Educational Games to Help Children in Communication Interactively. In *Computational Science and Its Applications – ICCSA 2016*, Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Ana Maria A.C. Rocha, Carmelo M. Torre, David Taniar, Bernady O. Apduhan, Elena Stankova, and Shangguang Wang (Eds.). Springer International Publishing, Cham, 465–473.

Ahmad F. Klaib, Nawaf O. Alsrehin, Wasen Y. Melhem, Haneen O. Bashtawi, and Aws A. Magableh. 2021. Eye tracking algorithms, techniques, tools, and applications with an emphasis on machine learning and Internet of Things technologies. *Expert Systems with Applications* 166 (2021), 114037. https://doi.org/10.1016/j.eswa.2020.114037

Per Ola Kristensson, James Lilley, Rolf Black, and Annalu Waller. 2020. A design engineering approach for quantitatively exploring context-aware sentence retrieval for nonspeaking individuals with motor disabilities. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–11.

Yun Li, Xiang Chen, Jianxun Tian, Xu Zhang, Kongqiao Wang, and Jihai Yang. 2010. Automatic Recognition of Sign Language Subwords Based on Portable Accelerometer and EMG Sensors. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction* (Beijing, China) *(ICMI-MLMI '10)*. Association for Computing Machinery, New York, NY, USA, Article 17, 7 pages. https://doi.org/10.1145/1891903.1891926

Runpeng Liu, Joey Salisbury, Arshya Vahabzadeh, and Ned Sahin. 2017. Feasibility of an Autism-Focused Augmented Reality Smartglasses System for Social Communication and Behavioral Coaching. *Frontiers in Pediatrics* 5 (06 2017). https://doi.org/10.3389/fped.2017.00145

Leo Marco and Giovanni Maria Farinella. 2018. *Computer vision for assistive healthcare*. Academic Press.

Don Mcmahon, David Cihak, Rachel Wright, and Sherry Bell. 2015. Augmented Reality for Teaching Science Vocabulary to Postsecondary Education Students With Intellectual Disabilities and Autism. *Journal of Research on Technology in Education* 48 (12 2015), 1–19. https://doi.org/10.1080/15391523.2015.1103149

Claire Mitchell, Gabriel Cler, Susan Fager, Paola Contessa, Serge Roy, Gianluca De Luca, Joshua Kline, and Jennifer Vojtech. 2022. Ability-Based Keyboards for Augmentative and Alternative Communication: Understanding How Individuals' Movement Patterns Translate to More Efficient Keyboards: Methods to Generate Keyboards Tailored to User-Specific Motor Abilities. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, Article 412, 7 pages. https://doi.org/10.1145/3491101.3519845

Stylianos Mystakidis, Athanasios Christopoulos, and Nikolaos Pellas. 2022. A systematic mapping review of augmented reality applications to support STEM learning in higher education. *Education and Information Technologies* 27, 2 (2022), 1883–1927.

Christopher S. Norrie, Annalu Waller, and Elizabeth F. S. Hannah. 2021. Establishing Context: AAC Device Adoption and Support in a Special-Education Setting. *ACM Trans. Comput.-Hum. Interact.* 28, 2, Article 13 (apr 2021), 30 pages. https://doi.org/10.1145/3446205

Mmachi God'sglory Obiorah, Anne Marie Marie Piper, and Michael Horn. 2021. Designing AACs for People with Aphasia Dining in Restaurants. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.

Sethuraman Panchanathan, Meredith Moore, Hemanth Venkateswara, Shayok Chakraborty, and Troy McDaniel. 2018. Computer Vision for Augmentative and Alternative Communication. In *Computer Vision for Assistive Healthcare*. Elsevier, 211–248.

Carole Plasson, Dominique Cunin, Yann Laurillau, and Laurence Nigay. 2020. 3D tabletop ar: a comparison of mid-air, touch and touch+ mid-air interaction. In *Proceedings of the International Conference on Advanced Visual Interfaces*. 1–5.

G. Porter, J. Kirkland, and Spastic Society of Victoria. 1995. *Integrating Augmentative and Alternative Communication Into Group Programs: Utilising the Principles of Conductive Education*. Spastic Society of Victoria. https://books.google.com/books?id=weYGPQAACAAJ

António Ramires Fernandes, Camilla Almeida da Silva, and Ana Grohmann. 2014. Assisting Speech Therapy for Autism Spectrum Disorders with an Augmented Reality Application. *ICEIS 2014 - Proceedings of the 16th International Conference on Enterprise Information Systems* 3.

Vidas Raudonis, Rimvydas Simutis, and Gintautas Narvydas. 2009. Discrete eye tracking for medical applications. In *2009 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies*. 1–6. https://doi.org/10.1109/ISABEL.2009.5373675

Philipp A Rauschnabel, Reto Felix, and Chris Hinsch. 2019. Augmented reality marketing: How mobile AR-apps can improve brands through inspiration. *Journal of Retailing and Consumer Services* 49 (2019), 43–53.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 779–788.

Ana Patrícia Rocha, Afonso Guimarães, Ilídio C. Oliveira, Fábio Nunes, José Maria Fernandes, Miguel Oliveira e Silva, Samuel Silva, and António Teixeira. 2022. Toward Supporting Communication for People with Aphasia: The In-Bed Scenario. In *Adjunct Publication of the 24th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Vancouver, BC, Canada) *(MobileHCI '22)*. Association for Computing Machinery, New York, NY, USA, Article 8, 4 pages. https://doi.org/10.1145/3528575.3551431

Omer Berat Sezer, Erdogan Dogdu, and Ahmet Murat Ozbayoglu. 2018. Context-Aware Computing, Learning, and Big Data in Internet of Things: A Survey. *IEEE Internet of Things Journal* 5, 1 (2018), 1–27. https://doi.org/10.1109/JIOT.2017.2773600

Howard C Shane, Sarah Blackstone, Gregg Vanderheiden, Michael Williams, and Frank DeRuyter. 2012. Using AAC technology to access the world. *Assistive technology* 24, 1 (2012), 3–13.

Junxiao Shen, Boyin Yang, John J Dudley, and Per Ola Kristensson. 2022. KWickChat: A Multi-Turn Dialogue System for AAC Using Context-Aware Sentence Generation by Bag-of-Keywords. In *27th International Conference on Intelligent User Interfaces* (Helsinki, Finland) *(IUI '22)*. Association for Computing Machinery, New York, NY, USA, 853–867. https://doi.org/10.1145/3490099.3511145

Kiley Sobel, Alexander Fiannaca, Jon Campbell, Harish Kulkarni, Ann Paradiso, Ed Cutrell, and Meredith Ringel Morris. 2017. Exploring the Design Space of AAC Awareness Displays. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2890–2903.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Ruiliang Su, Xiang Chen, Shuai Cao, and Xu Zhang. 2016. Random forest-based recognition of isolated sign language subwords using data from accelerometers and surface electromyographic sensors. *Sensors* 16, 1 (2016), 100.

Taryadi and I Kurniawan. 2018. The improvement of autism spectrum disorders on children communication ability with PECS method Multimedia Augmented Reality-Based. *Journal of Physics: Conference Series* 947 (jan 2018), 012009. https://doi.org/10.1088/1742-6596/947/1/012009

Stavroula Tzima, Georgios Styliaras, and Athanasios Bassounas. 2019. Augmented reality applications in education: Teachers point of view. *Education Sciences* 9, 2 (2019), 99.

Wei Wang, Songgui Lei, Haiping Liu, Taojin Li, Jue Qu, and Ang Qiu. 2020. Augmented reality in maintenance training for military equipment. In *Journal of Physics: Conference Series*, Vol. 1626. IOP Publishing, 012184.

Lilian Wanzare, Alessandra Zarcone, Stefan Thater, and Manfred Pinkal. 2016. DeScript: A crowdsourced Corpus for the Acquisition of High-Quality Script Knowledge.

Shengjing Wei, Xiang Chen, Xidong Yang, Shuai Cao, and Xu Zhang. 2016. A component-based vocabulary-extensible sign language gesture recognition framework. *Sensors* 16, 4 (2016), 556.

Xiaoyi Zhang, Harish Kulkarni, and Meredith Ringel Morris. 2017. Smartphone-based gaze gesture communication for people with motor disabilities. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2878–2889.

Haifeng Zhao, Petra Karlsson, Omid Kavehei, and Alistair McEwan. 2021. Augmentative and Alternative Communication with Eye-gaze Technology and Augmented Reality: Reflections from Engineers, People with Cerebral Palsy and Caregivers. In *2021 IEEE Sensors*. 1–4. https://doi.org/10.1109/SENSORS47087.2021.9639819

Yuhang Zhao, Elizabeth Kupferstein, Hathaitorn Rojnirun, Leah Findlater, and Shiri Azenkot. 2020. The effectiveness of visual and audio wayfinding guidance on smartglasses for people with low vision. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–14.

Cheng Zheng, Caowei Zhang, Xuan Li, Fan Zhang, Bing Li, Chuqi Tang, Cheng Yao, Ting Zhang, and Fangtian Ying. 2017. KinToon: A Kinect Facial Projector for Communication Enhancement for ASD Children. In *Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 201–203. https://doi.org/10.1145/3131785.3131813