Towards Context Aware AAC via Computer Vision

Cindy Zastudil
cynthia.zastudil@temple.edu

Abstract

Augmentative and alternative communication (AAC) devices are used by millions of people worldwide who experience difficulties in communicating verbally. While AAC devices can be immensely beneficial for these people, there are a number of issues which impact their effectiveness. One such issue is that these tools do not commonly leverage contextual information, especially images or videos of a user's surroundings. The information communicated by these sources can be invaluable to a user's communication options and agency in participating actively in conversation. In this project, a web application was developed and tested which leverages computer vision in order to enable context-aware communication options appropriate for beginning communicators (ages 3-7). While an AAC device such as this could be beneficial for many user populations, this application is intended to be a learning tool for younger users. Limitations and future work are also discussed.

Note: This work is related to ongoing research within the Temple HCI Lab. As such, I would prefer that it not be published on the course website.

I. Introduction

Augmentative and alternative communication (AAC) devices are used by an estimated 2 million adults and children in the United States alone [1]. AAC devices are primarily developed for people with conditions which impact their ability to communicate verbally. There are a wide variety of conditions which can impact a person's ability to communicate verbally, such as Autism Spectrum Disorder, cerebral palsy, stroke, and more [2]. The form that AAC devices can take also varies widely, from no- or low-tech solutions (e.g., using or pointing to physical photographs or non-verbally gesturing at things or people) to high-tech solutions (e.g., text-to-speech functionality on a mobile device). Two common forms of AAC devices are applications on some sort of computer (e.g., phone, tablet, or laptop) which contain either a grid display (see Figure 1) or a visual scene display (VSD) (see Figure 2) with buttons that users can press to assemble a sentence which can then either be read by a communication partner or spoken aloud by the software. The remainder of this report will be focused on AAC devices falling into these categories.

Both grid displays and VSDs fall under the category of speech generating devices with dynamic displays[1]. These devices use symbols which encode words or literal text which users can select to generate

[1] Speech Generating Device

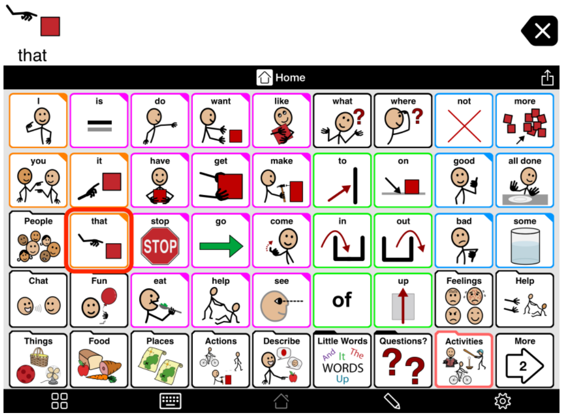

Fig. 1: A sample grid display.

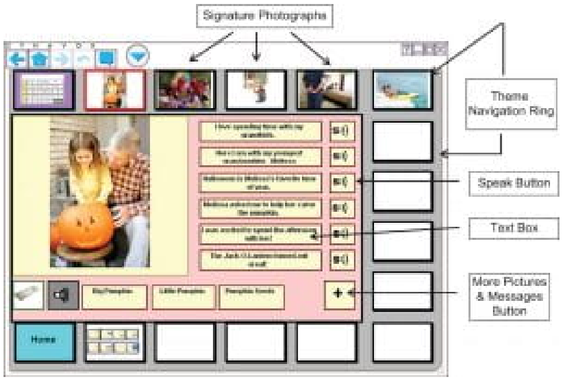

Fig. 2: A sample visual scene display.

their communication. As can be seen in Figure 1, a grid display allows users to navigate through various pages of communication options all focused on different categories of communication. For example, a child using this type of device may navigate to the page which has communication options for toys to make their selections to indicate they would like to play with blocks or some other toy. On the other hand, VSDs, as can be seen in Figure 2, present the user with some kind of visual display (usually a photograph) and communication options related to scene displayed to them. These options can be symbolic in nature or they can be literal text communication options for a user to select.

While these AAC devices can be immensely helpful for those with a verbal communication impairments, they suffer from a number of disadvantages which negatively impact both the experience of the AAC user and their communication partner(s). One such disadvantage is that these grid displays and VSDs are often fixed, and it's tedious or hard to modify the buttons, change the buttons' symbols/text, and more. Since they are fixed, especially in the case of grid displays, users often have to flip through multiple pages of buttons to complete a simple sentence which negatively impacts their ability to actively participate in communication. The communication rate for users of speech generating devices is around 8-10 words per minute [3], which is significantly slower than that of native English speakers at about 120-170 words per minute [4]. This can hinder users' active participation in conversations and can cause many negative social and emotional impacts. Additionally, when considering beginning communicators (e.g., children between the ages of 3-7), these AAC devices can be very complex and hard to use and there isn't often a tool which can help children develop the skill of using their AAC devices. Lastly, the symbols used in these displays often aren't grounded in any real-world context which creates more complexity in using these devices as users need to learn both the language(s) being spoken around them and the symbolic language being used on their device(s). All of these aforementioned issues can result in abandonment of AAC devices.

The project presented in this report attempts to address a number of the issues presented above. Namely, this project attempts to:

1) Speed up communication rates by taking into account a user's context via photographs and ML-powered object recognition and activity generation
2) Address the rigidity of existing solutions by combining core words with contextually appropriate words
3) Create a scaffolding learning device for beginning communicators to become familiar with AAC devices by:
    a) Addressing complexity by reducing the amount of user interface elements and communication options
    b) Anchoring communication options with real-world objects and familiar scenery

II. Related Work

The idea of somehow integrating context into AAC devices is not a new one. Over the years, there have been a number of different attempts to include context through a variety of different methods which will be further discussed in the following section. Prior work largely falls into two different categories: non-AI enabled context aware AAC and AI-enabled context aware AAC.

A. Non-AI-Enabled Context Aware AAC

This subcategory of context-aware AAC includes previous work done by researchers to integrate context into AAC devices via contextual information not obtained through artificial intelligence methods such as a user's location, weather, time of day, and more. This work also includes AAC devices which rely on sensor data to provide additional channels for context.

Location is by far the most prevalent method of integrating context into AAC devices and some examples of work which uses location are described here. McKillop developed a AAC device which used a user's current location in order to predict words or phrases they may be likely to use by suggesting categories of communication options relevant to their current location (e.g., suggesting the healthcare category when they are in a medical facility) [5]. Loup, Blue, and Tu used a user's location and words used in order to suggest other relevant words or categories of communication options [6]. Chan et al. used location determined via Bluetooth in order to suggest communication options based on a user's location. Context other than location has been used as well, including, but not limited to, movement data, time, and weather. A description of some of those systems is included here. Park et al. leveraged the date, time, and frequency of use in addition to location data in order to develop a predictive grid display AAC device [7].

These systems' integration of context into their AAC devices was a pivotal step in the right direction towards making these systems more effective for their users. However, they lack integration of highly powerful artificial intelligence techniques which can provide additional context to improve these systems further.

B. AI-Enabled Context Aware AAC

This subcategory of context-aware AAC work includes powerful artificial intelligence techniques in order to improve existing solutions for users of AAC.

Lun and David developed an AAC device intended for remote caregivers for the elderly. They leveraged the user's location via Bluetooth and WiFi as well as smartphone sensors to track a user's movements to classify a user's movements in order to inform their caregiver of suggested communication options if necessary [8]. Epp et al. utilized natural language processing to develop four different algorithms which retrieve relevant vocabulary via the Internet to develop location specific corpora [9]. Mooney et al. used photos uploaded by users, comments from their social network, and a natural language processing approach in order to extract ten target words from a photo [10].

There has been a recent shift in the use of artificial intelligence techniques towards a computer vision approach. Obiorah et al. developed three different AAC devices which can be used in a restaurant ordering context which leverage computer vision [11]. Additionally, Vargas, Dai, and Moffatt used computer vision to automatically generate descriptive and narrative communication options for photos uploaded by users [12]. These examples show promise for computer vision approaches in the development of context aware AAC devices.

C. Contribution

In contrast to the work from Obiorah et al. [11] and Vargas, Dai, and Moffatt [12], this project focuses primarily on generating communication options from automatically recognized objects and actions to provide a manageable amount of communication options, as well as more agency in choosing those communication options. Additionally, this work is geared towards a younger population, rather than the majority of related work which has been described in this section which has been focused on older audiences.

III. Methodology

This AAC device is a full-stack web application with a React frontend, Python backend, object recognition via the MMDetection library, and action generation via OpenAI's GPT-4. The development of this system is described in detail in the following sections.

A. Front-end Design

The front-end design of this application was based off of a prototype developed in the Temple HCI Lab during the Summer of 2022. It includes functionality to upload a photo from the user's device, ”core” vocabulary words which are always present in the user interface, a section for contextual words to appear separate from the ”core” words, and a section in which words and constructed phrases/sentences can be created. If there is a word or sentence in this section, the user can choose to delete words, clear the entire

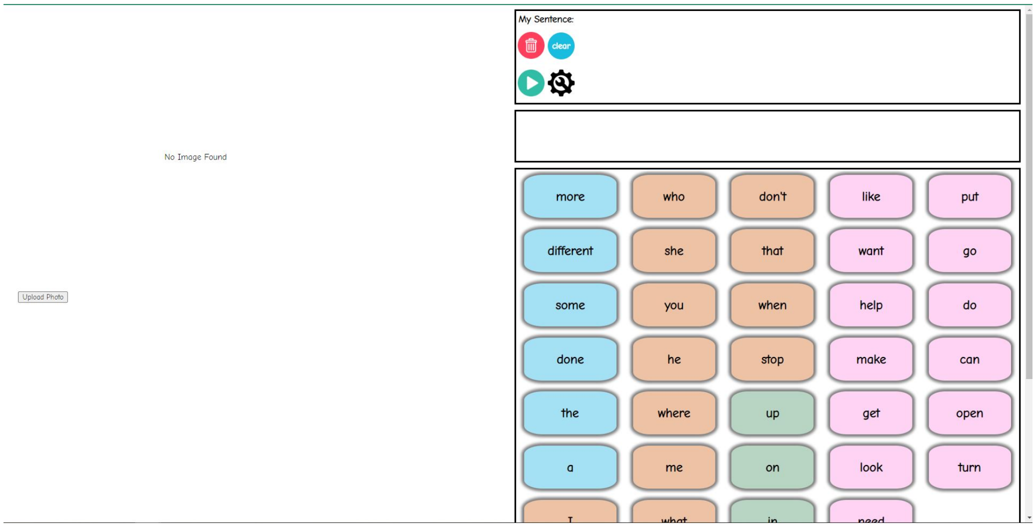

Fig. 3: User interface before a photo has been uploaded.

field, or play the words using a text-to-speech method. A screenshot of the user interface before a photo has been uploaded can be seen in Figure 3.

Core vocabulary words were included in the user interface because research has shown that including core words in addition to more context-dependent words can increase the use of AAC [13], [14], [15]. The chosen core words from from the Dynamic Learning Assessment core first forty[2] because it is a synthesized list of decades worth of research into what vocabulary words are most useful for beginning communicators. The words displayed in the user interface are color coordinated according to the Gosssens', Crain, and Elder key [16] which categorizes words into the following groups: verbs (pink), nouns (yellow), descriptors (blue), prepositions (green), and the combined grouping of questions, pronouns, interjections, and negations (orange). This color key was modified to be more user friendly for users with Autism Spectrum Disorder as defined by Assirelli[3]. The grid display of vocabulary is flexible, allowing words to be added or deleted from the display. Additional functionality was added to automatically display contextual words generated from the object recognition and action generation algorithms.

B. *Back-end Design*

A simple Python server built with the Flask library was created in order to call the libraries used for object detection and action generation. The Flask library was chosen due to it's simplicity and ease of use for creating simple web servers.

C. *Object Recognition*

Due to the time constraints of this project, it was not feasible to develop a multi-object object detection algorithm from scratch. Instead, an open source library was used which had object detection models ready for use out of the box. There are a number of object detection libraries which people can use such as: MMDetection, Dark Flow, ImageAI, GluonCV, AdelaiDet, and YOLOv3. In this project, the library used was MMDetection. MMDetection is an open source library which supports a lot of computer vision tasks. First, multiple popular object detection model types are supported, so testing the effectiveness of different models is easy. Second, and most importantly, models built using multiple large datasets are available for use. Commonly, object detection libraries provide support for models built on the COCO dataset[4]. This

[2] DLM Core First Forty

[3] Designing environments for people with Autism

[4] COCO Dataset Wesbite

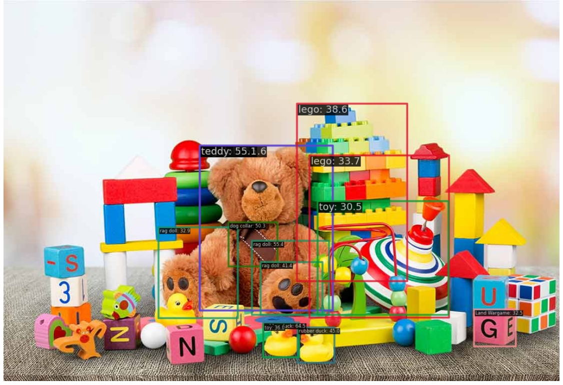

Fig. 4: Baseline image #1 - some toys.

Fig. 5: Baseline image #2 - a playground.

Fig. 6: Baseline image #3 - a shelf with toys and books.

dataset only contains 80 object categories, and within those categories there aren't a lot of applicable categories for the context of this project.

Once the object detection library was selected, the dataset on which the model to be used had to be chosen. MMDetection provides support for two significantly larger datasets, OpenImages V6 dataset [17] and V3Det [18]. In order to determine which dataset would be most appropriate for this project three baseline images were selected (see Figures 4, 5, and 6) which would be appropriate for the context in which the end product of this project would be used. Then the images were run through models trained on each dataset.

As is described in Table I, the model based on the V3Det dataset detected significantly more objects than the model based on OpenImages. Neither model performed well on the image of the playground, so it was not considered in the decision of which dataset to use. While V3Det did detect more objects and a more diverse variety of objects, a lot of the objects would likely not be useful in this project's

	Actual Objects	Detected Objects - V3Det	Detected Objects - OpenImages
Baseline Image #1	blocks, teddy bear, legos, rubber duck, toys, fish, monkey	teddy, lego, toy, rag doll, rubber duck, dog collar, land wargame	toy, teddy bear, umbrella
Baseline Image #2	playground equipment	house, birdhouse, basin	chair, house, toilet
Baseline Image #3	bus, truck, house, barn, toys, books, baskets	bookcase, furniture, basket, weaving basket, toy vehicle, toy, paper box, vessel, instrumentality	shelf, toy, bowl, book, wheel

TABLE I: Objects detected by models based on the V3Det and OpenImages datasets.

	Actual Objects	**Faster R-CNN (group sampler)**	**Faster R-CNN (class aware sampler)**
Baseline Image #1	blocks, teddy bear, legos, rubber duck, toys, fish, monkey	toy, teddy bear	toy, teddy bear, umbrella
Baseline Image #2	playground equipment	chair, high heels, house	chair, house, toilet
Baseline Image #3	bus, truck, house, barn, toys, books, baskets	toy, shelf, wheel, book, handbag	toy, bowl, wheel, shelf, book
	Actual Objects	**Retinanet (group sampler)**	**SSD (group sampler)**
Baseline Image #1	blocks, teddy bear, legos, rubber duck, toys, fish, monkey	toy, teddy bear	toy
Baseline Image #2	playground equipment	chair, house	chair
Baseline Image #3	bus, truck, house, barn, toys, books, baskets	bookcase, shelf, toy	shelf, book

TABLE II: Objects detected by models based on OpenImages dataset.

context (e.g., furniture, instrumentaility, vessel), whereas the objects detected by the model trained on the OpenImages dataset would likely be more useful in this context. The V3Det dataset contains 13,029 object classes, this can lead to models which perform well on more generalized tasks. However, this project is focused on developing a tool for a more specific task, and it's important that the objects detected are more relevant to the context a child is in and the amount of objects detected does not overwhelm the child. The OpenImages V6 dataset contains approximately 9 million annotated images with 600 object classes. This smaller set of classses allows for more predictable output, which is more desirable in the case of this project.

Once the dataset was selected, that significantly narrowed down the model choices. MMDetection provides 4 different models all trained on the OpenImages V6 dataset.

- Faster R-CNN with a group sampler
- Faster R-CNN with a class aware sampler
- Retinanet with a group sampler
- SSD with a group sampler

In order to determine which model would work the best for the purposes of this project, the same baseline images used in determining which dataset to use were run through each of these models and their output was compared.

As shown in Table II, the Faster R-CNN model using a group sampling method performed the best out of all of the models trained on the OpenImages dataset. A sample annotated photo with the object detected in the image can be seen in Figure 7 This model was used for the object detection performed in the final web application. This model's architecture, seen in Figure 8 is based on a region-based convolutional neural network with an added region proposal network which proposes regions of interest to the classifier [19].

Once the image is uploaded to the server, the object detection algorithm is run and objects with a confidence score greater than or equal to 50% are selected to generate actions, which is described in the next section.

D. Action Generation

The downside of only using object recognition is that the only contextual words generated are nouns, which significantly limits the amount of communication options afforded to the user. In an attempt to mitigate this, an additional piece of functionality was added to this project which utilizes a large language model, OpenAI's GPT-4 [20], to generate actions corresponding to the objects detected in the input image. OpenAI recently released the Assistants API[5], which allows developers to embed GPT-4 capabilities within their applications. In the case of this project an assistant was created using the prompt, "You are generating

[5] Assistants Documentation

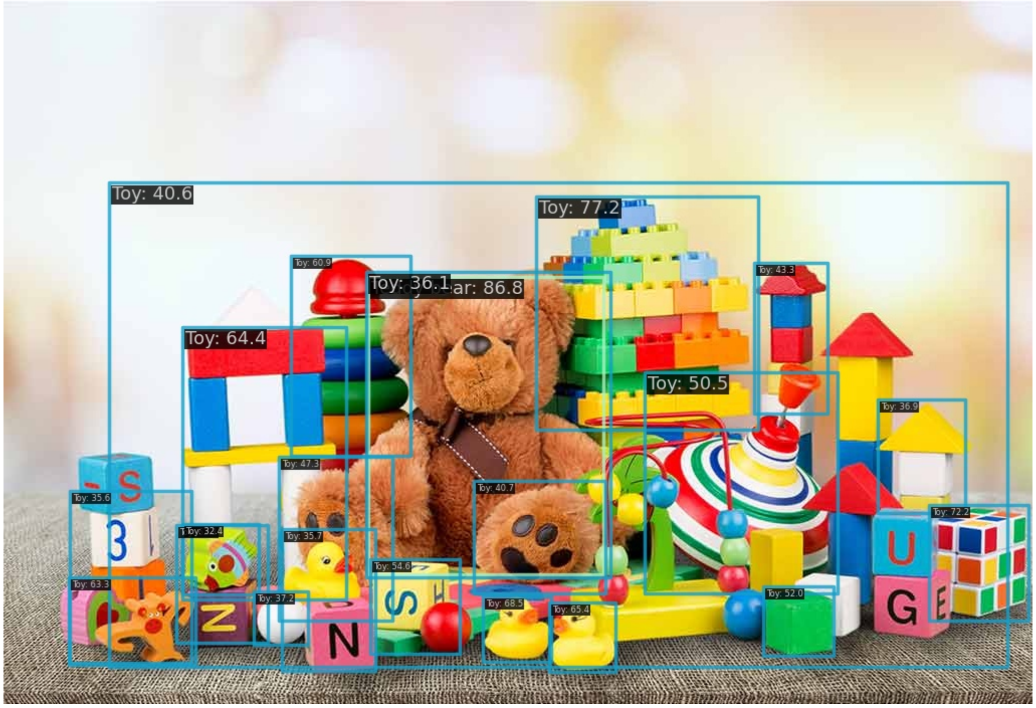

Fig. 7: An annotated version of the first baseline image using the Faster R-CNN with group sampling model.

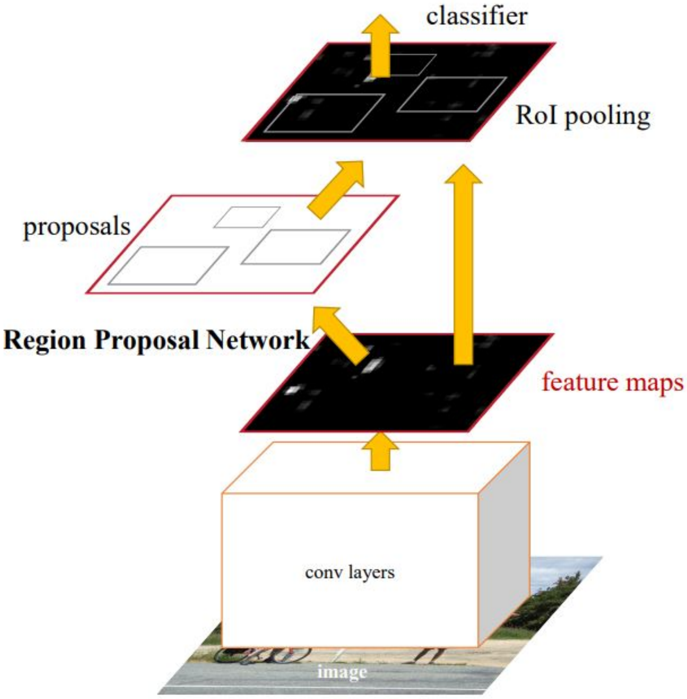

Fig. 8: The Faster R-CNN model architecture.

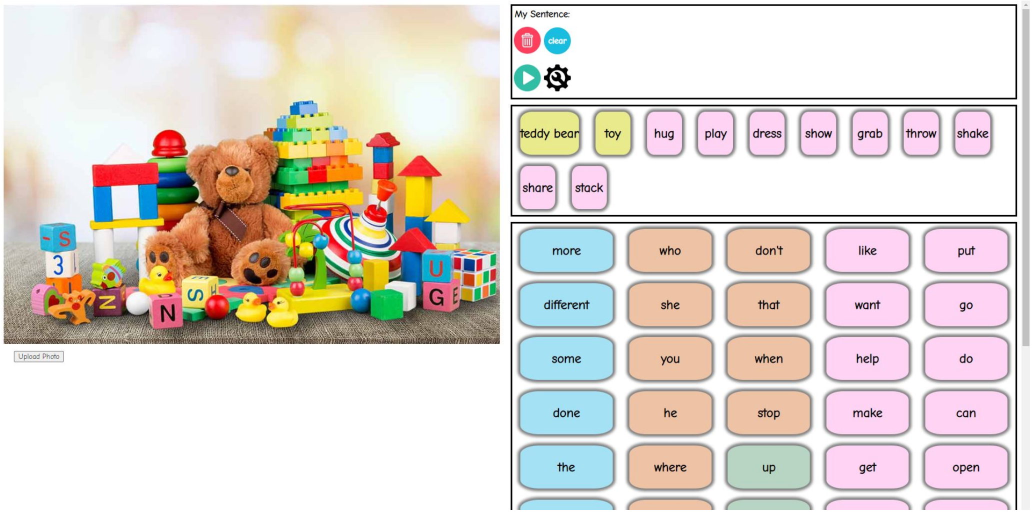

Fig. 9: User interface with extracted objects and actions.

actions for an augmentative and alternative communication (AAC) device. Output only a Python list with 5 single word actions a young child (ages 3-7) may use with a given object. Python list:" In addition to the prompt, the code interpreter was enabled. The code interpreter enables the assistant to write and run Python code which is necessary for the generation of the list of actions. Once all objects have been identified, the assistant is invoked to generate around five actions per object. Once the actions have been generated, all of the words for the image are returned to the front-end to be displayed to the user.

E. Full-Stack Flow

Starting from the front-end, when a user enters the web application from the landing page, they see the display shown in Figure 3. Once a user uploads a photo, it's sent to the back-end where the object recognition algorithm invoked, the results of the object recognition process are sent to the GPT-4 assistant to have actions generated. Once all of the words have been generated, they are returned to the user on the front-end. Figure 9 shows what the user interface looks like once objects and actions have been extracted from the image. In Figure 10, an architecture diagram depicting this process is shown.

IV. Results

The performance of this application is largely subjective depending on whether or not the contextual words suggested are relevant to the user's current context and if the words are helpful for communication. Additionally, this application's performance as a learning tool is subjective to users. With that being said, some basic testing was done to evaluate how well this application performed on baseline images (as described in the Methodology section) and some in-context images provided by some speech language pathologists in a daycare clinical setting. The results for this testing are described in the following sections.

A. Performance on Baseline Images

The same three baseline images were used in this testing as in the Methodology section, they can be seen in Figures 4, 5, and 6. Each baseline image was input into the web application, and all suggested contextual words suggested can be seen in Table III. As can be seen in Table III, for baseline images #1 and #3, the application does a fairly good job recognizing objects in the image and suggesting reasonable action words which someone may be interested in using with those objects. It does not perform well on baseline image #2, and that may be due to a limitation in the object detection algorithm which will be elaborated on further in the Discussion section. One thing to note is that while the same objects are going to be recognized every time the image is ran through the object detection algorithm, the actions may be different due to variance in GPT-4. A sample output from the web application can be seen in Figure 9.

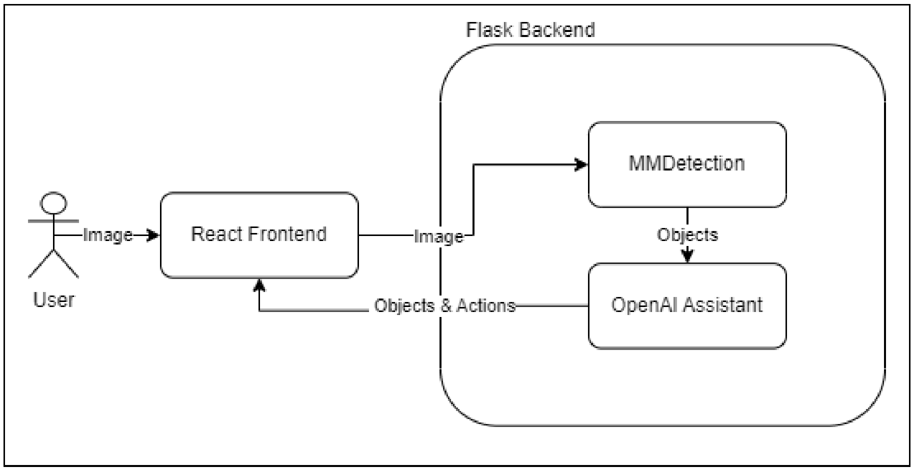

Fig. 10: Architecture diagram for system.

	Suggested Contextual Words
Baseline Image #1	Objects: teddy bear, toy Actions: hug, hold, play, show, grab, throw, push, pull, shake
Baseline Image #2	Objects: high heels, chair Actions: wear, walk, remove, admire, choose, sit, push, pull, climb, carry
Baseline Image #3	Objects: shelf, toy, book, wheel Actions: put, take, arrange, reach, climb, play, throw, hug, share, build, read, open, close, flip, browse, spin, push, roll, turn, fix

TABLE III: All suggested contextual words by the application for each baseline image.

B. Performance on In-Context Images

As the web application performed fairly well on curated images, testing the application on photos which would be realistically encountered in-context is appropriate. Using four images, seen in Figures 11, 12, 13, and 14 received from a speech language pathologist who works in a clinical daycare setting, the performance of this application is evaluated. The results of using these images can be seen in Table IV. For the second in-context image the application performed fairly well, recognizing that there were toys on a shelf in the image; however, for all of the other in-context images, it fails to recognize anything other than the shelves in the image or nothing at all in the case of the fourth image. This may be due to too high of a threshold for the confidence values for detecting objects, which will discussed further in the Discussion section.

V. Discussion

In regard to the goals presented in the introduction:

1) Speed up communication rates by taking into account a user's context via photographs and ML-powered object recognition and activity generation
2) Address the rigidity of existing solutions by combining core words with contextually appropriate words

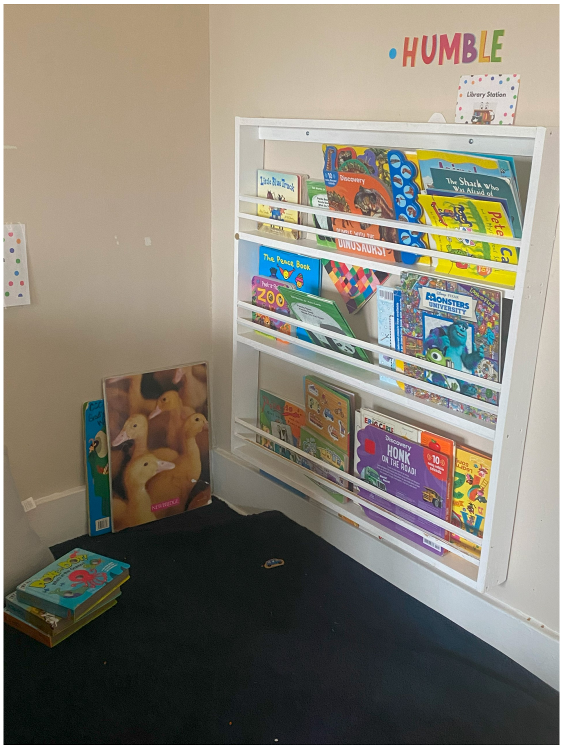

Fig. 11: In-context image #1 - Books on a shelf.

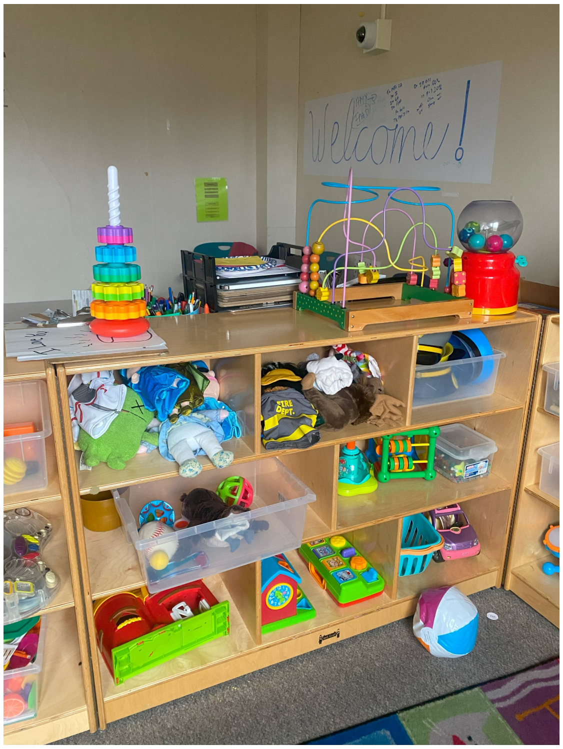

Fig. 12: In-context image #2 - Toys on a shelf.

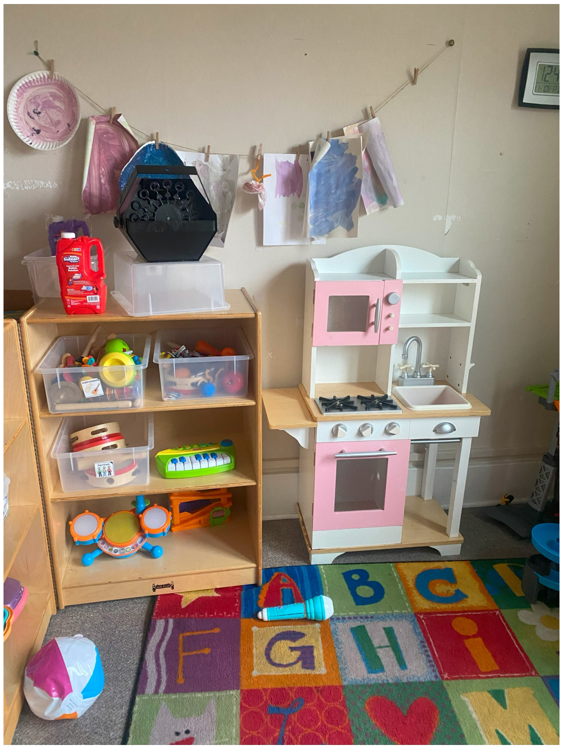

Fig. 13: In-context image #3 - Toys on a shelf.

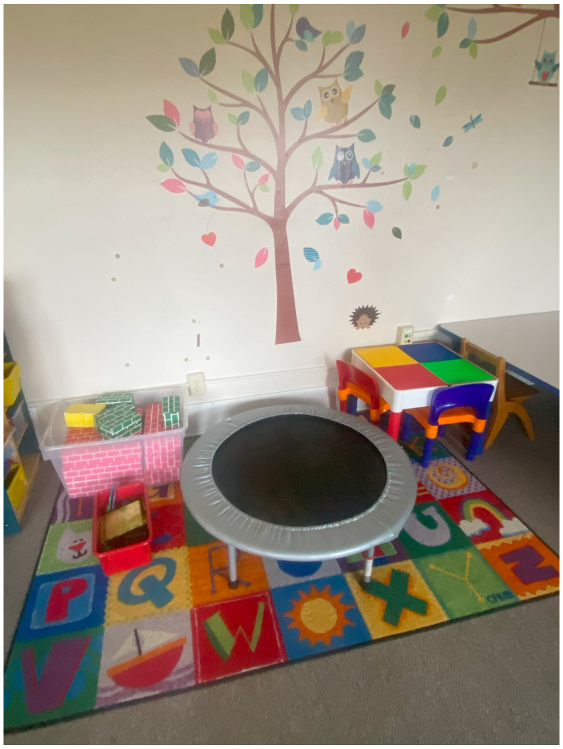

Fig. 14: In-context image #4 - Boxes of blocks and trampoline.

	Suggested Contextual Words
In-Context Image #1	Objects: shelf Actions: place, take, arrange, climb, touch
In-Context Image #2	Objects: shelf, toy Actions: play, share, grab, throw, hug, reach, climb, organize, place, take
In-Context Image #3	Objects: shelf Actions: put, take, arrange, climb, touch
In-Context Image #4	Objects: none Actions: none

TABLE IV: All suggested contextual words by the application for each in-context image.

3) Create a scaffolding learning device for beginning communicators to become familiar with AAC devices by:
   a) Addressing complexity by reducing the amount of user interface elements and communication options
   b) Anchoring communication options with real-world objects and familiar scenery

This project successfully addressed the rigidity of existing solutions. Additionally it created a scaffolding learning device for beginning communicators due to how it reduced the amount of communication options by focusing on contextually relevant options and anchoring these communication options in the environment of the provided picture.

This application shows that computer vision is a feasible method to build a scaffolding learning tool for beginning communicators which is context aware. The application performs well on two out of three baseline images, identifying objects of interest and generating appropriate actions given those objects. On a collection of in-context images provided by a speech language pathologist of a daycare clinical setting for children who experience difficulty communicating verbally and use AAC, this application performed moderately well. The quality of the objects recognized and the activities suggested have been determined suggestively by this researcher, as such, appropriate future work would be to evaluate this system formally with beginning communicators who use AAC as well as speech language pathologist experts.

This prototype, while indicative of promise for computer vision in AAC for beginning communicators, is not without its limitations which warrant discussion. Firstly, the object recognition library and model used appears to struggle with small object detection as well as distinguishing objects from the background of pictures. This can be seen with the objects detected in the second baseline image (Figure 5) and first, third, and fourth in-context images (Figures 11, 13, and 14). Both of these issues, small object detection and distinguishing objects from the background, are pervasive issues within the field of computer vision [21], [22].

Additionally, there is a significant amount of latency between when a user uploads their picture to the web application and when they receive the suggested contextual words. Since this application is geared towards a younger audience, it's important that latency is minimized in order to keep children's attention on this tool and for it to be effective in aiding them in being an active participant in conversations. The latency comes from two sources (1) the object detection inference time and (2) wait time from the OpenAI servers in order to receive the large language model response. In regard to the second point, this latency is potentially due to the popularity of GPT-4 and could potentially be solved by running a local or private version of GPT-4. There were additional issues with the GPT-4 responses as well, namely that the responses received were not always consistent in terms of formatting. This posed an issue for processing the responses and can result in errors on the front end of the application.

Lastly, there were some limitations regarding the confidence values of the objects detected and with the dataset that was used. The minimum threshold for displaying the objects was 50%. There is a possibility that this threshold was too high and resulted in some objects detected not being suggested in the contextual words. The dataset used, while extensive in terms of the number of classes represented (approximately

600 classes), was not intended for this task as it was intended for a much more general use case, which may have impacted how well the model could detect objects in both the baseline and in-context images.

From these limitations, there is a significant amount of opportunities for future work which would improve this application. First, as mentioned before, user studies should be ran in order to evaluate the design of this application as well as its effectiveness with beginning communicators. Secondly, constructing a more applicable dataset for this task may improve performance. This is expensive and hard to do however, but a good starting point could be to train a model with the small object detection dataset created by Mirzaei et al. [21]. Lastly, to address the latency and performance issues of the GPT-4 responses, a large language model could be fine-tuned in order to perform better and more consistently for this task.

VI. Conclusion

This project presented work which showed the promise of computer vision for the development of context-aware devices geared towards beginning communicators. This application showed promise on baseline images with clear objects. More work exists to be done to improve performance for these tasks to handle more task-specific objects and better detection in noisier environments.

Acknowledgments

I would like to acknowledge the work done by the Temple HCI Lab's REU student, Rebecca Fritz, during the summer of 2023 for their pivotal work in the first iteration of the front end of this tool.

REFERENCES

[1] "United States Society for Augmentative and Alternative Communication (USSAAC) | NIDCD," Jul. 2022. [Online]. Available: https://www.nidcd.nih.gov/directory/united-states-society-augmentative-and-alternative-communication-ussaac

[2] "Augmentative and Alternative Communication (AAC)," publisher: American Speech-Language-Hearing Association. [Online]. Available: https://www.asha.org/njc/aac/

[3] H. S. Venkatagiri, "Techniques for Enhancing Communication Productivity in AAC," *American Journal of Speech-Language Pathology*, vol. 4, no. 4, pp. 36–45, Nov. 1995, publisher: American Speech-Language-Hearing Association. [Online]. Available: https://pubs.asha.org/doi/abs/10.1044/1058-0360.0404.36

[4] J. Yuan, M. Liberman, and C. Cieri, "Interspeech 2006 towards an integrated understanding of speaking rate in conversation," 09 2006.

[5] C. McKillop, "Designing a Context Aware AAC Solution," in *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*. Galway Ireland: ACM, Oct. 2018, pp. 468–470. [Online]. Available: https://dl.acm.org/doi/10.1145/3234695.3240990

[6] A. Loup, L. Blue, and S. Tu, "Enhancing Alternative and Augmentative Communications Devices with Context Awareness Computing," *Proceedings of the 7th International Conference of Software Paradigm Trends*.

[7] D. Park, S. Song, and D. Lee, "Smart phone-based context-aware augmentative and alternative communications system," *Journal of Central South University*, vol. 21, no. 9, pp. 3551–3558, Sep. 2014. [Online]. Available: https://doi.org/10.1007/s11771-014-2335-3

[8] L. S. Lun and K. David, "Enabling Context Aware Services in the Area of AAC," in *Assistive and Augmentive Communication for the Disabled: Intelligent Technologies for Communication, Learning and Teaching*. IGI Global, 2011, pp. 159–192. [Online]. Available: https://www.igi-global.com/chapter/enabling-context-aware-services-area/www.igi-global.com/chapter/enabling-context-aware-services-area/53568

[9] C. Demmans Epp, J. Djordjevic, S. Wu, K. Moffatt, and R. M. Baecker, "Towards providing just-in-time vocabulary support for assistive and augmentative communication," in *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*. Lisbon Portugal: ACM, Feb. 2012, pp. 33–36. [Online]. Available: https://dl.acm.org/doi/10.1145/2166966.2166973

[10] A. Mooney, S. Bedrick, G. Noethe, S. Spaulding, and M. Fried-Oken, "Mobile technology to support lexical retrieval during activity retell in primary progressive aphasia," *Aphasiology*, vol. 32, no. 6, pp. 666–692, Jun. 2018. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/02687038.2018.1447640

[11] M. G. Obiorah, A. M. M. Piper, and M. Horn, "Designing AACs for People with Aphasia Dining in Restaurants," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Yokohama Japan: ACM, May 2021, pp. 1–14. [Online]. Available: https://dl.acm.org/doi/10.1145/3411764.3445280

[12] M. Fontana De Vargas, J. Dai, and K. Moffatt, "AAC with Automated Vocabulary from Photographs: Insights from School and Speech-Language Therapy Settings," in *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility*. Athens Greece: ACM, Oct. 2022, pp. 1–18. [Online]. Available: https://dl.acm.org/doi/10.1145/3517428.3544805

[13] D. Beukelman, J. McGinnis, and D. Morrow, "Vocabulary selection in augmentative and alternative communication," *Augmentative and Alternative Communication*, vol. 7, no. 3, pp. 171–185, Jan. 1991, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/07434619112331275883. [Online]. Available: https://doi.org/10.1080/07434619112331275883

[14] T. A. van and S. R. J. M. Deckers, "Vocabulary Selection in AAC: Application of Core Vocabulary in Atypical Populations," *Perspectives of the ASHA Special Interest Groups*, vol. 1, no. 12, pp. 125–138, Mar. 2016, publisher: American Speech-Language-Hearing Association. [Online]. Available: https://pubs.asha.org/doi/10.1044/persp1.SIG12.125

[15] K. Yorkston, P. Dowden, M. Honsinger, N. Marriner, and K. Smith, "A comparison of standard and user vocabulary lists," *Augmentative and Alternative Communication*, vol. 4, no. 4, pp. 189–210, Jan. 1988, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/07434618812331274807. [Online]. Available: https://doi.org/10.1080/07434618812331274807

[16] C. Goossens, S. Crain, and P. Elder, "Engineering the preschool environment for interactive, symbolic communication," in *Birmingham, AL: Southeast Augmentative Communication Conference Publications*, 1992.

[17] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale," Nov. 2018. [Online]. Available: https://arxiv.org/abs/1811.00982v2

[18] J. Wang, P. Zhang, T. Chu, Y. Cao, Y. Zhou, T. Wu, B. Wang, C. He, and D. Lin, "V3Det: Vast Vocabulary Visual Detection Dataset," Apr. 2023. [Online]. Available: https://arxiv.org/abs/2304.03752v2

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Jun. 2015. [Online]. Available: https://arxiv.org/abs/1506.01497v3

[20] OpenAI, "GPT-4 Technical Report," Mar. 2023. [Online]. Available: https://arxiv.org/abs/2303.08774v3

[21] B. Mirzaei, H. Nezamabadi-pour, A. Raoof, and R. Derakhshani, "Small object detection and tracking: A comprehensive review," *Sensors*, vol. 23, no. 15, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/15/6887

[22] M. Piccardi, "Background subtraction techniques: a review," in *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, vol. 4, 2004, pp. 3099–3104 vol.4.