Gator: Developing a Customized Navigational Assistance App for Individuals With Blindness or Visual Impairments

Riya Sahu • University of Bristol Immersive Lab • Laidlaw Scholars Program (MIT)

INTRODUCTION

Navigating environments that prioritize visual communication, particularly with signage or images, can pose a significant challenge for blind and visually impaired individuals. Many existing navigation aids rely on Internet connectivity for more advanced AI detection models or connection to live volunteers. This motivates the need for a navigation tool that can reach users with limited access to the Internet or specific operating systems. A proposed software to achieve this is Gator, an application that aims to provide an easily customizable navigation assistant that can operate entirely locally on a mobile device. In order to allow future developers to build upon or customize Gator's work, the app remains open source.

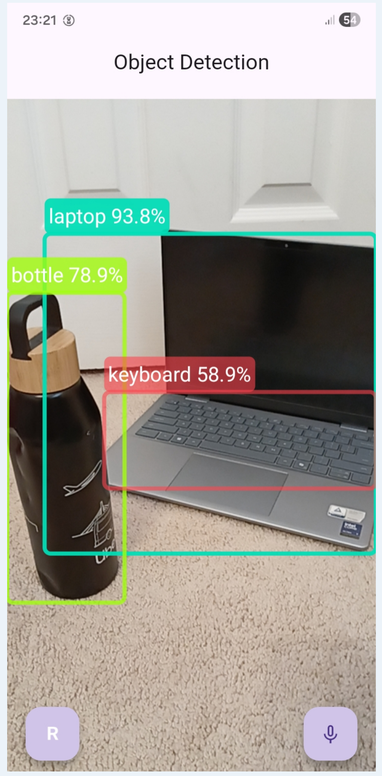
METHODS

Gator is a mobile and web application, programmed with Flutter and incorporating YOLO and Google MLKit models for detection. Utilizing the device's built-in microphone, speaker, and camera, it searches for user-specified visual targets in real time. It operates in text and object detection modes, each with their own settings. A microphone button enables speech-based commands from users, as shown in the table to the right. The context refresh button lets users start new detection sessions. These refreshes allow the user to adjust the camera position relative to a detected object or phrase in a feedback loop, or simply to focus elsewhere. With each command or detection, the app provides speech feedback to confirm that requests have been executed or detections have been made.

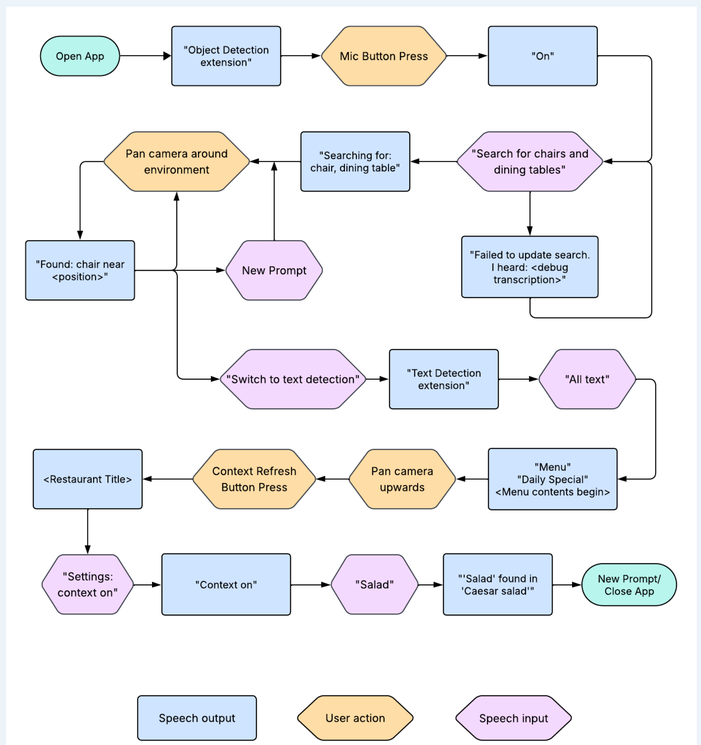
USER FEEDBACK

Volunteer test users tried Gator in a variety of environments, and they provided feedback on each iteration of the app. This feedback led to the addition of the context, substring, and echo settings. The context refreshing button, along with resetting announcements while the user activated the mic button, were also included to give the user a sense of selective focus. Users preferred to give natural language commands, and requested the option to use both audio and gestures, depending on the environment's surrounding noise level.

APP STRUCTURE



The screen-reader-navigable visual interface of Gator.



Example of user interaction with Gator during app testing: locating and learning about a restaurant nearby

ACTION	COMMAND
Detect Object	Any speech + <objects>
Detect Text	<exact phrase>
Switch Modes	Any speech + <mode>
Modify Setting	“Settings <setting> [on/off]”
List Settings	“Settings report”

SETTING	USAGE
Search	Allow detection and announcement
Position	Screen region where detection is
Color	Color information for objects
Substring	Allow matches that *contain* target text
Context	Announce line in which text was detected
Echo	Repeat user input after failure

Recognized speech-based commands in Gator along with available settings for customizing object and text detection modes.

DISCUSSION & LIMITATIONS

A majority of user feedback and suggested updates seemed to center around the idea of providing agency to the user: both in the settings they could customize and in the ability to selectively ignore or request new information. This suggests that maximizing the choices that Gator provides, without overwhelming the user, gives the best chance of successfully adapting to individual needs. Moreover, the tendency towards natural language suggests that providing less structure in commands will encourage users to interact more with Gator. Due to constraints on obtaining volunteer test users, it is difficult to conclusively decide that a design choice will be most beneficial for the majority of users. Publishing Gator to app stores may allow a wider range of users to provide feedback; however, this can raise consistency issues. In order to ensure that Gator can provide a seamless experience for users while also remaining responsive to their needs, a balance will be investigated between earlier release and more stability in updates.

FUTURE CONSIDERATIONS

In order to accommodate the differences in environments that Gator is used for, it may be beneficial to include different environment modes, each with their own specialized detections, or to allow users to upload their own datasets and models to the app. To ensure a more robust performance even in louder environments, future versions of Gator will likely include a gesture-based system for updating settings, in addition to the current audio method. Natural language command processing is planned in the next development, most likely utilizing a lightweight model for function calling to allow conversational language.

ACKNOWLEDGEMENTS

Thank you to Dr. Kyle Keane, who supervised this research project, for providing thoughtful design and software advice along with insight into Gator's mission for visually impaired community members. Thank you also to the researchers of the University of Bristol Immersive Lab for technical guidance and user testing, and to the Laidlaw Scholars Program and MIT MISTI for enabling this opportunity.