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

Riya Sahu
University of Bristol Immersive Lab
Laidlaw Scholars Program (MIT)
Supervised by Dr. Kyle Keane

September 1, 2026

Abstract

Navigation of unfamiliar spaces can pose an extra barrier to individuals with blindness or other visual impairments, as many of our current systems depend heavily on visual cues. To address this challenge, various assistive technologies for non-visual navigation have emerged. However, popular solutions often either require Internet access, are supported on limited mobile operating systems, or both. To enable accessibility for a wider range of individuals, the Gator app aims to provide a cross-platform, open-source navigation aid that functions without Internet connection. Gator is fully audio-navigable and operates from a mobile phone's camera and microphone. The two modes, object detection and text detection, track user-defined targets from the video feed in real time. Both modes allow the user to customize the information they receive, including settings on color, position, surrounding context, or audio debugging information. Volunteer users demonstrated a preference for customizability during testing, favoring natural language inputs over structured speech commands in the app, and preferring to change between strictness levels in the text detection mode. In addition, users expressed a desire to control the rate of information received, motivating the inclusion of context refreshing and interruptible announcements within the app. These added features allowed users to rapidly narrow down objects or text from the entire frame to only those of interest. In determining the utility of such features and preferred layouts, Gator not only functions as a standalone app, but as a publicly available starting point for developers to collaborate and continue broadening the reach of these accessibility solutions.

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

Introduction

A common challenge for blind and visually impaired individuals is the navigation of spaces that prioritize visual information, such as written signs or images (Messaoudi et al., 2022). Developing navigation technologies allows people with visual impairments to more safely interact with such environments, often described as a key factor of independent living (Kuriakose et al., 2022; Casanova et al., 2025). As the number of individuals with visual impairments is projected to increase by nearly 50% from 2020 in under 30 years, designing assistive technologies for non-visual navigation becomes essential for ensuring that public spaces remain accessible (Burton et al., 2021; Bourne et al., 2017).

One of the most widely used applications on the current market is Be My Eyes, a live assistance app where volunteers use speech to aid blind users in tasks, given a video feed (Senjam et al., 2021). While it has proven useful for many as a technology to help carry out everyday tasks involving identification, review of users' experience with the app reveals that fewer interviewed individuals employed it for navigation (Laura & Pudrianisa, 2024; Avila et al., 2016). Reasoning included a desire to maintain the user's privacy or insufficient location-specific context for the volunteer (Avila et al., 2016). Alternative solutions in the development stage currently include the OOrion and STEP Navigation apps for iOS, both of which explicitly focus on the task of navigation (Rhouma, 2026).

While both apps offer promising options for a target consumer base of iOS users with reliable Internet access, neither STEP Navigation nor OOrion is currently supported on Android, nor are there clear signs of open-source code (OOrion, 2025). User reviews for OOrion on AppleVis–a Be My Eyes partner company–also note that some detection features only work when generative AI features are enabled, requiring internet access (OOrion, 2024). Additionally, existing digital solutions have sparked privacy concerns over how blind users' data is preserved, either due to the unpredictability of a live volunteer, or the possibility of data being uploaded to the host and used to train AI models (Stangl et al., 2022). Moreover, many current navigation assistance apps do not meet user expectations when it comes to indoor navigation in particular (Bleau et al., 2026).

Finally, in many countries, a majority of individuals with visual impairments reside in rural areas as opposed to urban, and there tends to be a much higher proportion of visually impaired individuals in low-income regions than higher-income regions (Vashistha & Anderson, 2016). A study on technology usage patterns of low-income individuals in India reveals that less than one third of the country's population has Internet access (Vashishta & Anderson, 2016). As a result, many blind users are likely to be unable to use existing popular navigation technologies, which often require an Internet connection.

Here, we demonstrate one example of a potential solution to minimize these gaps in the consumer base: the application NaviGator (Gator). The Gator software aims to address privacy concerns and unreliable Internet connectivity by running detection models locally. In addition, in order to maximize reach across device types and operating systems, the app is designed to run on multiple mobile and web-based platforms. Gator remains open-source for future developers to customize and build upon, and to allow transparency to users.

Methods

Logical Architecture

Gator is programmed using Google's Flutter SDK for cross-platform compatibility. It is primarily intended for iOS App Store and Google Play deployment, but it also functions on web browsers including Microsoft Edge and Google Chrome provided that the device has an accessible microphone and camera. The lightweight object and text detection models employed by Gator operate locally on the user's device. This method has the added benefit of more reliable navigation, even in areas with less cell coverage or Internet access. The aim with this framework is therefore to provide an app that is more adaptable to users worldwide.

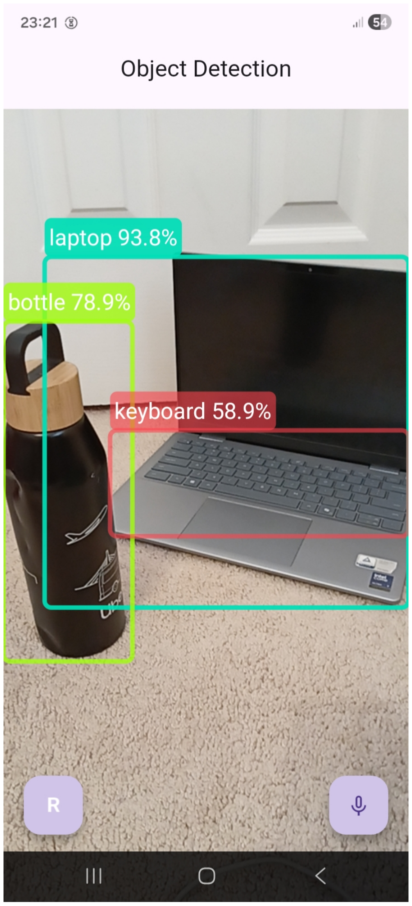

FIGURE 1. Standard Gator interface on a Samsung Galaxy A07.

Gator's structure places the user in one of two modes at any given time: object detection or text detection. The multi-class object detection mode utilizes the YOLO v11 Nano model from TensorFlow Lite to minimize computational requirements, maximizing the rate of video feed processing. The pre-trained version of this model recognizes 80 types of objects known as the COCO classes, which are common objects often used as a benchmark for assessing general object detection models (Singh et al., 2024). Among the classes are categories such as "car," "chair," and "laptop." As shown in Figure 1, which exemplifies a sighted user's experience of the interface, the app draws bounding boxes on the visual interface labeled with class and confidence score.

For the text detection mode, using Google MLKit's text detection library, Gator searches for a phrase within all text recognized in a camera frame. Depending on the settings the user adjusts, the app can announce the surrounding detected block of words as context or search within words for substrings, to prevent hyphenation or capitalization from causing detection misses. In both modes, the app announces each detection, by default with the name of the class or phrase and its position on the screen (specified by one of nine regions in a three-by-three grid from top left to bottom right).

User Interface

The user interface layout is highly minimized, designed to remain as compatible with screen readers as possible. The visual interface includes a live video feed reflecting the device's camera input, along with bounding boxes in object detection mode; however, this is a secondary aid and not an essential component for the app's function. There are three primary elements to be accessed via a screen reader. The first is a label at the top of the visual camera feed to keep a text and audio record of what mode the user is in. In addition, there are two buttons: a context refresh button, and a microphone control button. Both buttons are annotated for screen readers to describe their intended function.

The context refresh button allows the user to start a new detection session from the video feed frame immediately following the button press. This action erases all prior detections and begins announcing new objects or text given the previous prompt. The microphone control button switches between active listening and an "off" state. When active listening is on, all current or queued detections are cleared, similar to asking a new question in a conversation. It is then that the user can describe an action with specific speech commands, as outlined in Figure 2 along with available settings. During the "off" state of the microphone, detection is enabled again unless explicitly turned off by the user.

Source Code

The most current version of Gator, with the updates described in this paper, is publicly available on GitHub at https://github.com/riya-sahu/gator-summer-2026. Both the debug and release APK outputs are functional for Android use, with developer options enabled. This repository is forked from the University of Bristol Immersive Lab organization; the original repository is available at https://github.com/UoB-AT/nav-si.

FIGURE 2. User actions and togglable settings in Gator.

Action	User Command	App Response
Object Request	Any speech + <object>	"Searching for: <object>"
Text Request	<phrase or "all text">	"Searching for: <phrase>"
Switch Modes	Any speech + "[object/text] detection"	"Switching to <mode>. Please wait"
Modify Setting	"Setting(s) <setting> [on/off]"	"<setting> [on/off]"
List Settings	"Settings report"	<mode> <all settings/values>

Setting	Modes Present	Purpose
Search	Both	Allow active detection and announcements
Position	Both	Describe one of nine regions on the screen for each detection
Color	Object Only	Include color information in object detection announcements
Substring	Text Only	Allow any instance of the target phrase's characters as a detection
Context	Text Only	Provide the line of text in which the detection occurred
Echo	Both	Repeat the transcription of the user input if the command fails

Results

Iterative Design

Gator is a project in progress, and the past three months of development–the focus of this paper–built on top of an existing draft, developed by University of Bristol intern Kailey Epstein with the assistance of undergraduate students Dixant Pant and Juan Pablo Ruiz Partida. Prior to user testing, there were two prototypes: a version with an object detection controller to autonomously manage the video feed, and a version with a cleaner interface, only one microphone button, and frame-by-frame image processing run manually for object detection. The new version of Gator first merged the two prototypes so that the simplified, screen-reader accessible layout was preserved, while keeping an object detection controller in the code to automatically process the video feed.

Feedback on the merged version of the app was then given in two iterations. To review the app's effectiveness, two blind volunteers and two sighted volunteers were asked to locate a specific object as well as gain information about their general environment through the app alone. While initial targets were suggested, in order to best understand how users might interact with the app, no prompting or redirection was used. After each round of feedback, the app was updated with requested features or suggestions from test findings.

User Testing and Feedback

Much of the feedback for the first testing iteration focused on the addition of settings: users requested certain features to be togglable where they had initially been assumed. For example, the text detection mode was set such that only the exact transcription of the user's request would be announced if detected, with no additional surrounding context provided. So, if a user searched for a word such as "full", then a phrase such as "half-full" would not be detected. In addition, users found that they were unable to rapidly make or change requests due to the ongoing announcements from previous detections.

One of the main drivers of the second iteration of the app was therefore the idea of maximizing agency of the user. Choices that had previously been rigid as design decisions were converted to dynamic settings to accommodate a wider range of preferences. This led to the new iteration including the togglable settings of substring and context. When substring was enabled, any word or line of text containing the same string of characters as the transcription would pass as a detection. However, if the user wished to match the exact whitespace and hyphenation patterns of the text to their speech input, disabling the substring setting would enforce the stricter system with fewer detection hits. The context setting then allowed the user to hear the line in which the phrase was found, enabling them to understand the placement of the target word without having to search for all text indiscriminately.

Announcement interrupts were added into the audio system to prevent the issue of the app speaking over the user, allowing them to choose when to start a new detection request. To further customize detection reports, a new function was developed called context refreshing, activated by a separate button. This button would halt any announcements, clear all queued detections, and start from the new video feed, aiming to create a parallel between the app's selective focus and that created by human eyes focusing away from objects at will.

Feedback in the second iteration of user testing centered more around interactions with the existing functions of Gator, rather than requests to add new functionalities. One of the key noticings of both testing rounds was that users, regardless of visual or non-visual interaction with the app, instinctively used natural language commands at first. Even after being informed that the app would only recognize specific wordings as executable actions, users would provide full sentences as prompts, often in a conversational manner. Users then expressed some signs of difficulty in adapting to the rigid phrasing of the commands versus using natural language.

In addition, while test users appreciated the speech-based output of detection reports, they expressed less satisfaction with speech-based input through the microphone button. Users noted that loud environments interfered with command recognition. Two test users, who were existing users of native screen readers on their mobile devices, also noted that screen reader and app audio outputs often conflicted. These individuals expressed interest instead in a gesture recognition-based system for modifying settings, or a visual settings interface that could be navigated by screen reader.

In order to increase clarity in describing audio command issues to the user, the echo setting was included in the revised version. When a user enabled echo in Gator, the app would read aloud the speech-to-text transcription it received from the user upon encountering an error. Although this was not a full solution to the issue of re-issuing commands, or being unable to give commands at all in loud environments, the feature aimed to assist users with having their speech recognized by the speech-to-text model.

Figure 3 below demonstrates the typical flow of a new user's interaction with the most updated version of the app, using the example of searching for information about a restaurant.

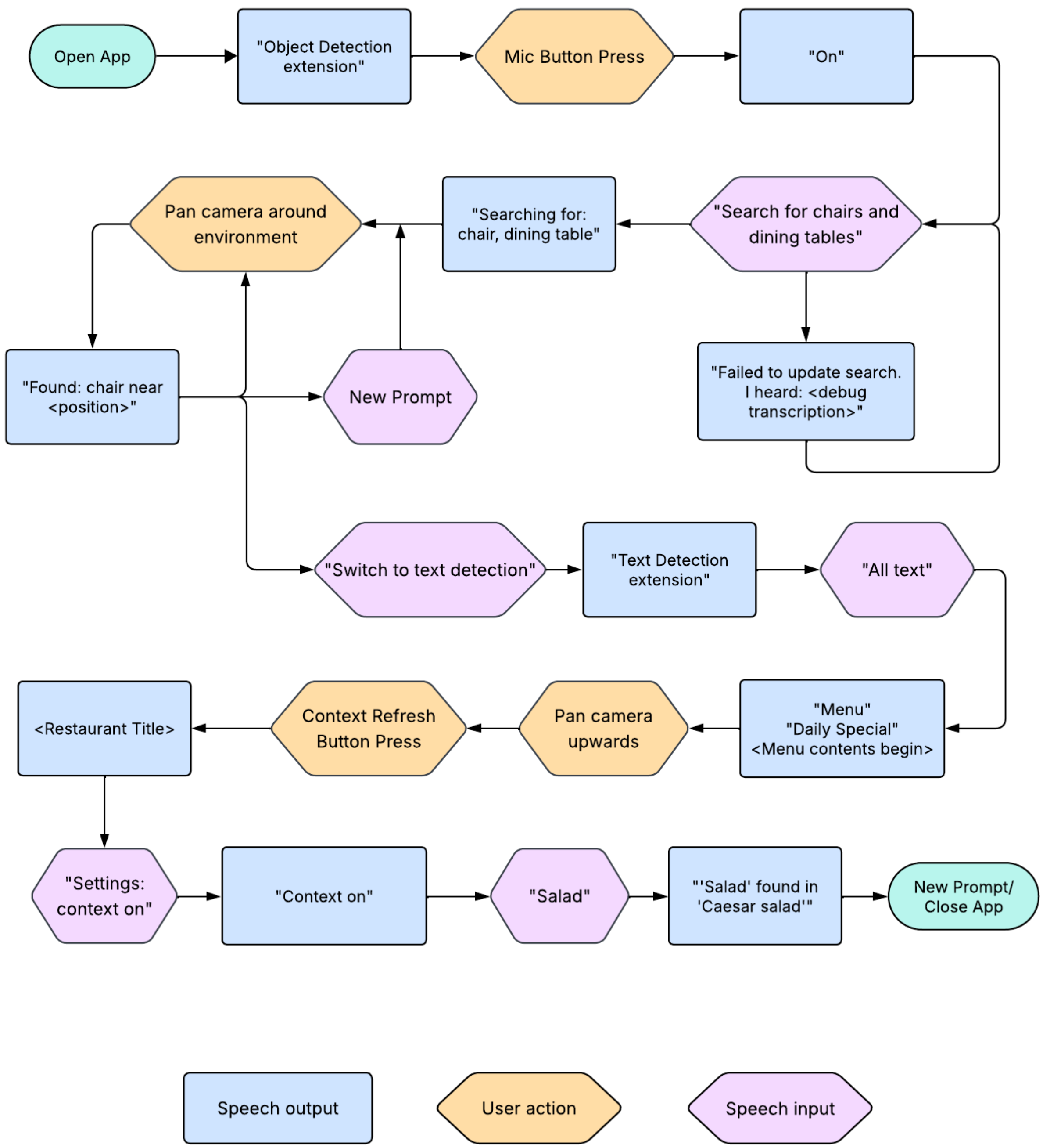

FIGURE 3. Approximate user interaction with Gator. Mic button presses and "On" announcements occur before every speech input, and mic button presses additionally occur after every input. However, for brevity, this has only been demonstrated once.

Discussion

Interpretations of Tests

The issue of selective context appeared in much of the feedback, which agrees with the common finding of the lack of an exact parallel between visual and audio figure-ground processing. In the visual medium, our ability to maintain foreground versus peripheral vision provides a balance between focus and general context (Wagemans et al., 2012). However, our ability to make these figure-ground separations in hearing often depends heavily on active focus rather than passive distinction (Teki et al., 2019; Qiu & von der Heydt, 2005). This becomes a challenge when audio is a shared sense–multiple people speaking in a room do not have the same independence as multiple objects being visible, as the individual conversations may combine to a listener (Bronkhorst, 2015).

This issue presented itself to users while Gator was tested in loud environments as well as when users found that the screen reader and app output speech collided. The preference for gesture recognition is likely an attempt to minimize these conflicts, due to users experiencing difficulty distinguishing the foreground of the app from the background of environmental noise or screen reader annotations. To some extent, users' preference for speech interrupts with the mic and reset buttons may stem from this as well. These interrupts allow a user to quickly re-listen to a detection of interest, much like a visual user making a second observation of a sign if it was obstructed while they were reading. In this way, even when a user misses a detection due to background noise, it is possible to easily retrieve it.

The test interactions also suggested that users appreciate natural language, along with a conversational approach to navigation, rather than an unambiguous set of speech constructs for prompting searches and detections. This theme logically follows the preference for more customizable settings rather than concrete design choices, as found from the first round of feedback. Ultimately, user testing revealed that much of the feedback revolved around the idea of providing flexibility and choice to the user.

Limitations

A primary limitation during the iterative process was access to volunteers for testing. While the app remains unpublished, it is difficult to obtain a wider range of test consumers, especially users with visual impairments who are most likely to use the app in practice. This leads to the consideration: when should Gator be considered ready for deployment to public app stores? On one hand, it may be highly beneficial to release early, so that feedback can be received from a more diverse audience of users.

On the other hand, updates from this feedback are likely to become jarring for a regular user trying to familiarize themselves with Gator's structure. Even without layout changes, added functions are not always made apparent to users with visual impairments, leading to many users opting out of updates or preferring more stable apps (Bhowmick et al., 2025; Lee et al., 2024; Wentz & Lazar, 2016). This leads to a second issue of balancing continuity for ease of use with responsive updates that better suit users' needs. In addition to the changes that the updates themselves might present, there would be the necessity of communicating updates and new features to users in an accessible manner.

Lastly, although gesture recognition can improve the user's experience while inputting information, loud environments may still inhibit a person's ability to concentrate on app detections. The resulting limitation is that Gator may not yet provide as much assistance in particularly noise-dense environments, where it may otherwise have the most potential to aid, given that users can no longer rely on the text-to-speech functions.

Future Work

In this same light, since users will navigate a wide variety of different environments, there is a compelling argument for developing specific modes of object detection based on the type of environment. For example, navigating a grocery store may warrant a more specialized classification of produce items, while a street navigation mode may require a model more thoroughly trained on signage, construction barriers, or features of crossings and sidewalks. Taking this idea a step further for maximum customization, we may consider allowing the user to upload custom models or choose from a set of existing detection models to upload custom data sets to.

Many existing detection models and feature trackers are computationally expensive to include, either due to memory constraints or buffering (Wagner et al., 2010). Given that many specialized detection models compatible with mobile phones are developed for specific, non-navigational tasks, it may be difficult for users to find viable models (Ignatov et al., 2019; Miluzzo et al., 2010). However, usage of older, lightweight YOLO models for specialized tasks has proved adaptable to mobile phones in the past (Zeng et al., 2023; Deng et al., 2009). Further investigation would therefore be beneficial to determine if there is a set of alternative models that will preserve the real-time nature of Gator's detections.

In response to the second round of user testing, where gesture recognition was requested for settings, Gator will be equipped with a touch-activated settings menu. In order to remain accessible to users who do not use a screen reader or have a motor disability, it is likely that both audio and gesture-based controls will be retained with full functionality together.

Lastly, in order to allow for a more intuitive and conversational approach, Gator will likely be transitioning to accept unstructured commands, most likely with a small local model to map natural language to app functions. Existing function calling models, such as Google's Function Gemma or IBM's Granite-20B-FunctionCalling, have proven lightweight enough to run reliably on mobile phones (Kamath et al., 2025; Abdelaziz et al., 2024). More investigation on function calling models will then be beneficial to determine how to incorporate this mapping without compromising the real-time processing of the video feed.

Acknowledgements

First and foremost, I would like to thank my research supervisor this summer, Dr. Kyle Keane, without whose guidance I would not have been able to work with Gator, let alone build out the features of a new version. His mentorship allowed me to adopt the technical goals of Gator, but also to better understand the underlying principles of its development. I would also like to thank my colleagues in the University of Bristol Immersive Lab for volunteering their time to test Gator in its various stages of development, and for providing technical and moral support. Lastly, I am extending my gratitude to the Laidlaw Scholars Program and MIT MISTI Coordinators for providing me with the opportunity to learn from my talented colleagues in a new setting and contribute to a meaningful project these past three months.

References

Abdelaziz, I., Basu, K., Agarwal, M., Kumaravel, S., Stallone, M., Panda, R., Rizk, Y., Bhargav, G. S., Crouse, M., Gunasekara, C., et al. (2024). Granite-function calling model: Introducing function calling abilities via multi-task learning of granular tasks. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 1131–1139.

Avila, M., Wolf, K., Brock, A., & Henze, N. (2016). Remote assistance for blind users in daily life: A survey about be my eyes. *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. https://doi.org/10.1145/2910674.2935839

Bhowmick, H., Shrivastava, A., Ysp, S., Anand, S., & Chakraborty, D. (2025). Crippin'whatsapp's interaction design: Learnings from the blind/visually impaired users of india. *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility*, 1–16.

Bleau, M., Martiniello, N., Gingras-Royer, N., Tardif-Bernier, C., & Nemargut, J. P. (2026). Exploring the use of smartphone applications during navigation-based tasks for individuals who are blind or who have low vision: Future directions and priorities. *Disability and Rehabilitation: Assistive Technology*, *21*(2), 840–868.

Bourne, R. R., Flaxman, S. R., Braithwaite, T., Cicinelli, M. V., Das, A., Jonas, J. B., Keeffe, J., Kempen, J. H., Leasher, J., Limburg, H., et al. (2017). Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: A systematic review and meta-analysis. *The Lancet Global Health*, *5*(9), e888–e897.

Bronkhorst, A. W. (2015). The cocktail-party problem revisited: Early processing and selection of multi-talker speech. *Attention, Perception, & Psychophysics*, *77*(5), 1465–1487.

Burton, M. J., Ramke, J., Marques, A. P., Bourne, R. R., Congdon, N., Jones, I., Tong, B. A. A., Arunga, S., Bachani, D., Bascaran, C., et al. (2021). The lancet global health commission on global eye health: Vision beyond 2020. *The Lancet Global Health*, *9*(4), e489–e551.

Casanova, E., Guffanti, D., & Hidalgo, L. (2025). Technological advancements in human navigation for the visually impaired: A systematic review. *Sensors*, *25*(7), 2213.

Deng, J., Yang, C., Huang, K., Lei, L., Ye, J., Zeng, W., Zhang, J., Lan, Y., & Zhang, Y. (2023). Deep-learning-based rice disease and insect pest detection on a mobile phone. *Agronomy*, *13*(8), 2139.

Ignatov, A., Timofte, R., Chou, W., Wang, K., Wu, M., Hartley, T., & Van Gool, L. (2018). Ai benchmark: Running deep neural networks on android smartphones. *European Conference on Computer Vision*, 288–314.

Kuriakose, B., Shrestha, R., & Sandnes, F. E. (2022). Tools and technologies for blind and visually impaired navigation support: A review. *IETE Technical Review*, *39*(1), 3–18. https://doi.org/10.1080/02564602.2020.1819893

Laura, N., & Pudrianisa, S. L. G. (2024). The utilization of the be my eyes application as a communication medium for visually impaired individuals in pertuni yogyakarta. *Proceeding Of International Conference on Communication and Media Digital*, *1*(1), 234–248.

Lee, J. G., Lee, B., Choi, S., Seo, J., & Choe, E. K. (2024). Identify, adapt, persist: The journey of blind individuals with personal health technologies. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, *8*(2). https://doi.org/10.1145/3659585

Lu, H., Pan, W., Lane, N. D., Choudhury, T., & Campbell, A. T. (2009). Soundsense: Scalable sound sensing for people-centric applications on mobile phones. *Proceedings of the 7th international conference on Mobile systems, applications, and services*, 165–178.

Messaoudi, M. D., Menelas, B.-A. J., & Mcheick, H. (2022). Review of navigation assistive tools and technologies for the visually impaired. *Sensors*, *22*(20). https://doi.org/10.3390/s22207888

Miluzzo, E., Cornelius, C. T., Ramaswamy, A., Choudhury, T., Liu, Z., & Campbell, A. T. (2010). Darwin phones: The evolution of sensing and inference on mobile phones. *Proceedings of the 8th international conference on Mobile systems, applications, and services*, 5–20.

OOrion. (2024). *AppleVis*. https://www.applevis.com/apps/ios/utilities/oorion.

OOrion. (2025). *The OOrion App*. https://www.oorion.fr/en/the-app.

Qiu, F. T., & Von Der Heydt, R. (2005). Figure and ground in the visual cortex: V2 combines stereoscopic cues with gestalt rules. *Neuron*, *47*(1), 155–166.

Rhouma, R. B. (2026). Assistive mobile application for visually impaired based on real time object recognition using machine learning with voice feedback. *Lusophone Scientific Repository*. http://hdl.handle.net/10437.1/14841

Senjam, S. S., Manna, S., & Bascaran, C. (2021). Smartphones-based assistive technology: Accessibility features and apps for people with visual impairment, and its usage, challenges, and usability testing. *Clinical optometry*, 311–322.

Singh, S., Yadav, A., Jain, J., Shi, H., Johnson, J., & Desai, K. (2024). Benchmarking object detectors with coco: A new path forward. *European Conference on Computer Vision*, 279–295.

Stangl, A., Shiroma, K., Davis, N., Xie, B., Fleischmann, K. R., Findlater, L., & Gurari, D. (2022). Privacy concerns for visual assistance technologies. *ACM Trans. Access. Comput.*, *15*(2). https://doi.org/10.1145/3517384

Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., et al. (2025). Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Teki, S., Chait, M., Kumar, S., von Kriegstein, K., & Griffiths, T. D. (2011). Brain bases for auditory stimulus-driven figure–ground segregation. *The Journal of Neuroscience*, *31*(1), 164–171.

University of Bristol Immersive Lab. (2026a). Gator [computer software] [Forked from University of Bristol Immersive Lab's NAV-SI by Riya Sahu.]. https://github.com/riya-sahu/gator-summer-2026

University of Bristol Immersive Lab. (2026b). NAV-SI [Computer software]. https://github.com/UoB-AT/nav-si

Vashistha, A., & Anderson, R. (2016). Technology use and non-use by low-income blind people in india. *SIGACCESS Access. Comput.*, (116), 10–21. https://doi.org/10.1145/3023851.3023853

Wagemans, J., Elder, J. H., Kubovy, M., Palmer, S. E., Peterson, M. A., Singh, M., & Von der Heydt, R. (2012). A century of gestalt psychology in visual perception: I. perceptual grouping and figure–ground organization. *Psychological bulletin*, *138*(6), 1172.

Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., & Schmalstieg, D. (2010). Real-time detection and tracking for augmented reality on mobile phones. *IEEE Transactions on Visualization and Computer Graphics*, *16*(3), 355–368. https://doi.org/10.1109/TVCG.2009.99

Wentz, B., & Lazar, J. (2016). Exploring the impact of inaccessible redesign and updates. In *Designing around people: Cwuaat 2016* (pp. 3–12). Springer.

Zeng, T., Li, S., Song, Q., Zhong, F., & Wei, X. (2023). Lightweight tomato real-time detection method based on improved yolo and mobile deployment. *Computers and electronics in agriculture*, *205*, 107625.