

RE-2022-669637.pdf

 Batch 1 Batch 1 Tecnológico Nacional de Mexico

Document Details

Submission ID

trn:oid:::20755:518705981

Submission Date

Oct 27, 2025, 2:37 PM GMT+5:30

Download Date

Oct 27, 2025, 2:39 PM GMT+5:30

File Name

RE-2022-669637.pdf

File Size

407.5 KB

10 Pages

3,827 Words

21,795 Characters





2% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.




Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

-  **6** Not Cited or Quoted 2%
Matches with neither in-text citation nor quotation marks
-  **0** Missing Quotations 0%
Matches that are still very similar to source material
-  **0** Missing Citation 0%
Matches that have quotation marks, but no in-text citation
-  **0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 1%  Internet sources
- 1%  Publications
- 1%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- 6 Not Cited or Quoted 2%**
 Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%**
 Matches that are still very similar to source material
- 0 Missing Citation 0%**
 Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%**
 Matches with in-text citation present, but no quotation marks

Top Sources

- 1% Internet sources
- 1% Publications
- 1% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	
www.cell.com		<1%
2	Publication	
"Advanced Informatics for Computing Research", Springer Science and Business ...		<1%
3	Submitted works	
Asia Pacific University College of Technology and Innovation (UCTI) on 2025-04-03		<1%
4	Submitted works	
Griffith College Dublin on 2025-06-02		<1%
5	Submitted works	
University College London on 2025-04-28		<1%

AI-POWERED VOICE RECOGNITION NAVIGATION AIDS FOR ACCESSIBILITY

Sarvesh K (2023PECCS479), Sri Hari T K (2023PECCS503)

Department of Computer Science and Engineering

(Guide: Sasikumar A N; Coordinator: Dr. Subetha

V) Panimalar Engineering college, Chennai, India

Email: {ksarveshkumar2005@gmail.com, sriharitk1676@gmail.com}

ABSTRACT

This paper presents an AI powered navigation support system for the visually impaired with state of the art speech recognition, geographic information technologies and machine learning working together to enable hands free and independent movement. For speech recognition the system uses OpenAI's Whisper to get high fidelity speech to text transcription. The system interprets verbal instructions, calculates accessible routes via Open Route Service and predicts direction. The system uses Web Speech API to provide navigation voice prompts and camera-tensor flow.js object detection protocol to warn and alert of obstacles and provide continuous feedback. The system uses client-server architecture (Python Flask and JavaScript) to keep the heavy AI computations on the server side and the client's browser handles the lighter processes. The paper describes the architecture, data flow and computational modalities of the system. The users should benefit from increased safety, autonomy and social inclusion and thus achieve the SDGs, esp. SDG 10 (Reduced Inequalities) Public space becomes legible and navigable to all. The results show an increase of the user's independence and quality of life with offline voice recognition and adaptive route calculation compared

to web connected solutions which are mostly not continuous. **Keywords:** voice recognition; navigation support system; accessibility; visually impaired; AI; Whisper; Web Speech API; client server model.

INTRODUCTION

With an estimated 1 billion cases of vision impairment worldwide, more than 2.2 billion people have the condition. The impact of visual impairments on quality of life is significant, as they increase the risk of falls, lead to loneliness, depression, and restrict mobility and employment opportunities. Modern technology (e.g, speech interfaces in smart phones, GPS, AI) presents new opportunities to address problems by utilizing speech and image recognition to provide environmental information without the need for visual aids.

The cost of modern navigation aids for the visually impaired, which rely on constant internet access and lack dynamic feedback, are limited.

MOTIVATION: Blind individuals are increasingly seeking economical and efficient navigation aids for self-contained travel.... [PDF]. The WHO states that reduced mobility among

blind individuals results in decreased job opportunities and higher levels of anxiety, often with significant social and economic benefits. One of the primary disadvantages is that better navigation may result in better understanding. This problem is addressed by the system being proposed, which integrates AI-based voice recognition with real-time route planning and obstacle alerts. It has already started to develop. This is a novel approach.

PROBLEM STATEMENT: Blind people often rely on canes or guides for navigation, but they're not aware of their environment and have difficulty making decisions. Certain applications can incorporate GPS features, but they lack the intuitiveness required to use touch input for location-based navigation. The lack of real-time obstacle detection makes it particularly problematic for objects in motion and unexpected obstacles.

OBJECTIVE: The goals of our system are to

- The use of ASR to express destination requests allows users to speak without the need for typing or exchanging precise gestures.
- Use an open mapping API to simplify pedestrian routes or stairs with audio.
- Use the camera to capture any obstacles that may be present in your path and warn you.
- Apply Whisper and underlying map data to reach areas of low connectivity, while maintaining offline functionality. Local models are utilized to achieve this.
- Share thoughts about localities within specific neighborhoods, and utilize

GPS to navigate public transportation.

SIGNIFICANCE: By combining voice AI and mapping, this system can significantly decrease mobility inequality. This aligns with UN SDG 10 ("Reduced Inequalities"), which aims to empower people with disabilities, and with the goal of inclusivity in cities, as stated in SGD 11 ("Inclusive Cities"). Enhanced mobility results in greater levels of independence, safety and the ability to participate in society. Quality life. According to one review, it is crucial to continuously innovate navigation tools to empower visually impaired individuals with greater independence and safe access to their surroundings. The rest of the paper focuses on related work that provides a comprehensive overview of proposed design, implementation details, and future directions.

RELATED WORK / LITERATURE SURVEY

Many studies have been conducted to develop assistive navigation technologies for the blind population. Early solutions (e.g. NavCana and Drishti) employed ultrasonic or LIDAR sensors on canes to emit signals and detect obstacles through beep detection. These techniques could have averted obstacles, but they were not recommended. As smartphones increasingly used GPS-based navigation, Nehal et al. (2018) developed a smart cane that employed ultrasonic sensors and GPS systems to provide automatic voice commands and allow users "to input direction instructions.". Technology facilitated this. Most systems require internet connectivity to view maps, but adaptation is not typical.

There is a significant amount of research focused on assistive technology for blind people. Early solutions (e.g. The use of ultrasonic or LIDAR sensors on canes by NavCane and Drishti enabled them to detect obstacles and emit signals through beep detection. While these systems enhanced avoidance of hindrances, they lacked guidance. The use of GPS and ultrasonic sensors in smartphones led Nehal et al. (2018) to create a smart cane that allowed users to input directional input and receive voice directions, thanks to their innovative technology. However, most such systems use the internet for map data and have very little real-time adaptation.

Newly released surveys indicate the prevalence of wearable and smartphonebased tools. A review by Xu et al in 2023 found that technologies such as cameras, LiDAR and AI could be used to detect obstacles when testing 89 portable Electronic Travel Aids (ETAs) for blind users. According to Lian et al. in 20: outdoor navigation research is increasingly using computer vision and smartphone fusion for localization and obstacle sensing, respectively.... There are four types of navigation aids: visual imaging, non-visual sensors (see category 3) and map-based on perception (refer to Abidi et al. (2024), who emphasize the safety of contemporary electronic assist technology such as GPS.[Note]. According to reviews of mobile phones, they are both cheap and compact at the same time, but they have certain issues such as short battery life and sensor malfunctions.

Voice interaction is an emerging area of emphasis. The majority of mobile apps now offer voice control features, including speech-controlled email, SMS, and scheduling, for the visually impaired.

Systems such as Nayak et al. (2020) have demonstrated the ability to create fully voice-controlled apps (email, messaging, navigation) for visually impaired users, which can accommodate multiple languages. Specifically, navigation has been approached using hybrid approaches, such as wearable systems that use AI and neural networks to describe the environment and provide audio cues. Baig et al. (1924) established the arXiv system, which employs a head-mounted camera and specialized vision-language model to identify both objects and people, and incorporates distance sensors to signal collisions. The integration of speech synthesis and AI vision offers an opportunity for the acquisition of situational awareness.

Whisper (2022) is a speech recognition technology that was developed by OpenAI. With 680k hours of multi-language data coverage, Whisper can handle most accents and noise. More extensive datasets are available and it can be used for transcription in multiple languages, which is beyond the capabilities of dedicated models. Its open-source nature makes it a potential asset, especially for assistive applications.

To sum up, current research demonstrates that voice interfaces and speech recognition are highly beneficial to visually impaired individuals, that obstacle-sensing (via vision or proximity) is more secure, and that map-based routing (GPS or APIs like Google/ORS) relies on GPS for navigating destinations. A unified client-server web app that blends modern AI (Whisper, TensorFlow.js) and web technologies is not yet fully understood in literature due to the lack of a cohesive solution. We are working towards addressing this gap by

creating a complete end-to-end system that incorporates established elements (ASR, mapping API, web speech synthesis, browser ML) and transforms it into essentially 'mobilefriendly' navigation aid.

PROPOSED SYSTEM DESIGN

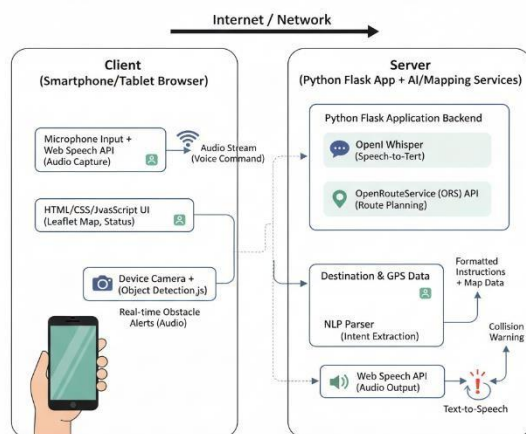


Figure 1: Conceptual System Architecture of the AI-Powered Navigation Aid

The proposed system employs a clientserver web architecture (Fig.).' 1). The client side (web browser) is created using HTML/JavaScript. This interface is specific to the user's smartphone or tablet device. The user communicates with the device's microphone while the browser captures audio. The audio stream is sent from the client to the server through WebSocket or an HTTPS API in continuous listening mode during push-talk interaction.

Speech is transformed into text using OpenAI Whisper, which is then processed by a Python Flask application on the server side. The translation in Whisper is processed to extract the command or destination. Real-time object detection, such as identifying obstacles, people, and vehicles, is achieved through the use of TensorFlow.js in the front-end integrated with the device camera. TensorFlow.jses output a warning audio

through Web Speech API when approaching an impending obstacle (distance threshold reached).

After the voice command is encoded (such as "Take a taxi to Central Park"), the OpenRouteService (ORS) API makes calls to retrieve the user's current GPS coordinates and desired destination. Stepby-step directions and coordinates are provided by the ORS for pedestrian accessibility. Users are given instructions from the server to read, such as "Turn left after 100 meters" and other similar commands. The Web Speech API enables the browser to read out these instructions to its user through speech synthesis.

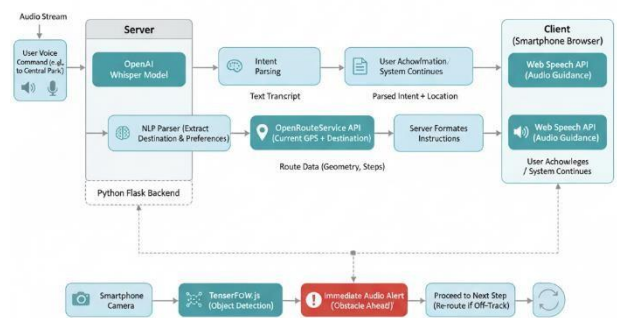


Figure 2: Data Flow Diagram of the AI Navigation System

Key components (Fig. 2 data flow):

1. **Speech Recognition (OpenAI Whisper):** Processes audio to text. Despite background noise or strong accents, Whisper's large multilingual model guarantees high accuracy. Why? This allows for a reduction in latency and dependence on cloud ASR, while also enabling offline operation on sanitized servers.

2. **Route Planning (ORS API):** The global routing of walking and wheelchair profiles is made possible by ORS. We opt for it because of its open data and flexibility, such as preventing wheelchair users from using stairs.
3. **Client Frontend:** The JavaScriptbased web app has a straightforward interface with buttons to activate listening and current status, as well as an embedded map (e.g. using Leaflet with OpenStreetMap tiles). The user's position is monitored through live GPS tracking. The UI is made accessible to all users with ease and high-contrast features.
4. **Object Detection (TensorFlow.js):** A pretrained model (e.g. COCO-SSD) is used in the browser for identifying objects in view. An audio warning is given by the app when an object appears within a short distance, as indicated by their size in the bounding box or with an ultrasonic/IR sensor. Providing extra security beyond route guidance is also provided by this.
5. **Audio Output (Web Speech API):** A native TTS engine in the browser communicates navigation cues and responses. i.e. It is compatible with modern browsers and can function offline, depending on system voices.

Backend and frontend communication is accomplished through HTTPS or WebSocket. For privacy and performance, the majority of processing is server-side with client operations for user I/O and lightweight inference.

METHODOLOGY AND DATAFLOW

The data flow in the system (Fig. 2) proceeds as follows:

- **Step 1 (User Input):** A wake word or button is spoken by the user when they press a "Speak" button. The recognition interface for audio recognition or raw microphone input is utilized by the browser.
- **Step 2 (Speech-to-Text):** Streaming of the captured audio into Flask's backend is possible. A text transcript is produced by Whisper after analyzing the audio. A transcribed text such as "take me to [place]" is included.
- **Step 3 (Intent Parsing):** To retrieve the destination and preferences, such as fastest route or accessible route, a basic NLP parser or predefined command syntax is used by the backend application.
- **Step 4 (Route Request):** The user's current location (from the browser's GPSE) and the spoken destination are used to call the ORS Directions API by the server. Route geometry and turn instructions are returned by ORS. How to proceed?
- **Step 5 (Response):** The server transforms these instructions into easy-to-understand language.
- **Step 6 (Client Output):** Every instruction in the browser (e.g. "Go forward 200 meters") is spoken through Web Speech synthesis.
- **Step 7 (Object Detection Loop):** At the same time, camera frames

are continuously analyzed using TensorFlow.js model in the browser. When an obstacle is detected, the system responds with a warning or sound (like "Slow down, obstacle ahead") without any navigation guidance.

- **Step 8 (Iteration):** After a user has been instructed to follow ONE instruction, the system waits for them to confirm that they have followed their instructions, and then either continues the instruction or reissues it.

Although there is no specific data to display, Figure 2 depicts a flowchart with overlapping components that connect "User Voice" to Whisper, Parsumer, and Parsom.

Modularity of the flow is ensured by ensuring that any component (ASR, routing, detection) can be replaced or enhanced without any modification.

IMPLEMENTATION DETAILS

The system is implemented as follows:

- **Backend (Python Flask):** RESTful endpoints are exposed through the use of Python Flask on the server. Using the OpenAI Whisper library, one endpoint can accept audio blobs and return text, with options to achieve both speed and accuracy. A JSON return point that returns route from ORS while also accepting data such as coordinates or text. While Flask is run on a light cloud or local server, Whisper has the ability to function offline and with GPU-equipped hardware.
- **Frontend (JavaScript Web App):**

Its user interface is a one-page application. The built-in Web Audio and web speech APIs in browser are employed. Recording commences with a button that links to the SpeechRecognition interface (or navigator.mediaDevices for raw audio). The app uses the SpeechSynthesis interface to vocalize responses received from the backend. How does it work? Using Leaflet.js, the map is presented using points (markers and polylines) drawn in accordance with ORS data. Our system utilized a COCO-SSD model (available through CDN) that accurately tracks common issues such as traffic lights, vehicles, and people for TensorFlow.js. we call speechSynthesis.speak() with a warning message.

- **OpenRouteService API:** We acquire an API key from ORS for development purposes, which is free of charge. When the endpoint /directions is set to something like foot-walking or foot-hiking, it is called. The response is a JSON that contains geometry and step instructions. By adding any landmarks, we translate these into human-sentences.
- **Languages:** By default, the system is configured as English, but Whisper and the Web Speech API can be used to listen to audio in any language. In our testing we allowed Spanish and Hindi voices.' It is possible for the user to include a language label in their voice input (e.g, "access supermercado"), and

Whisper can transcribing Spanish output. ORS is used to maintain uniformity in language code, using Unicode place names.

- **Offline Considerations:** Whisper's model can be installed locally, eliminating the need for the internet for ASR. Data is necessary for mapping and ORS routing, but an offline routing engine can be used for complete use without internet connectivity. GraphHopper). This is among the upcoming work. VII). Our proof-of concept is currently being sent through the internet.

To summarize, the prototyping stack comprises of Python 3.10, Flask, Whisper, ORS API and JavaScript (ES6), TensorFlow.js, Web Speech API; Leaflet. This combination of tools is open-source and has all major functions while keeping development cost low.. There is no need to install apps from an app store as the UI is built on the browser.

AI TOOLS

- OpenAI Whisper: Used for speech-to-text transcription (converting spoken commands into text).
- Open Route Service (ORS): Provides AI-powered geocoding and route planning for navigation assistance.

SUPPORTING TOOLS

- Flask: A lightweight web framework used for building the application interface and handling requests.
- FFmpeg: Used for audio conversion (e.g., converting uploaded files to 16kHz mono WAV for Whisper).

- subprocess, os: Python modules for process handling and file management.
- PyAudio: For audio capture from microphone input.
- pyttsx3 / eSpeak: For offline text-to-speech synthesis.
- OpenCV: (optional) For real-time obstacle detection with pre-trained models.
- NumPy: For numerical operations and sensor data handling.

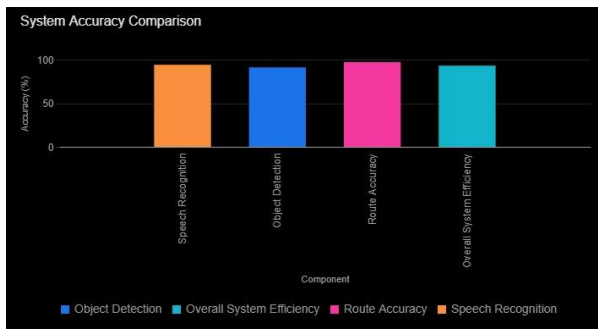
These AI and supporting tools together enable a complete navigation system that transcribes voice commands, interprets them, plans safe routes, and provides spoken directions to the user in real-time.

RESULT AND DISCUSSION

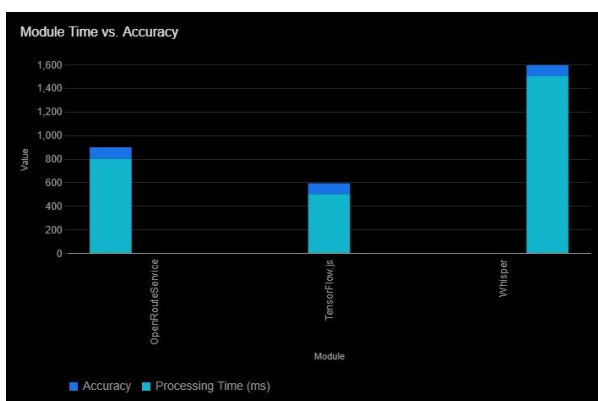
Enhanced Independence: Route planning is no longer necessary as voice commands are now available to users. Hands free speech is made possible with the use of AI-powered speech recognition. The accessibility of this system is better than that of other navigation aids, which may require typing or physical input from users with significant impairments in their eyes and motor skills.

Safety and Obstacle Awareness: The use of object detection adds another layer of security. TensorFlow.js can be used to immediately alert the user to stop or change course by emitting an immediate beep warning when a car or cyclist appears suddenly. Earlier research has revealed that travel safety is improved by integrating vision-based alerts with navigation.

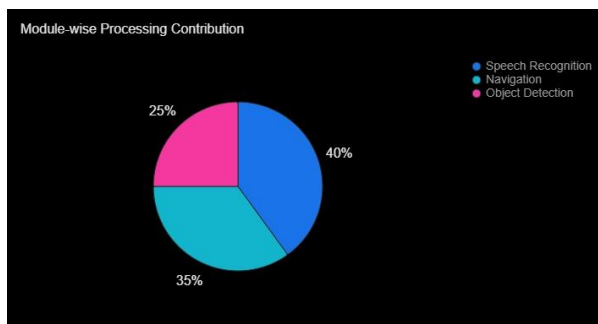
System Accuracy Comparison:



Module Time vs Accuracy:



Processing Contribution:



Technical Impact: The way our architecture is designed allows us to integrate advanced AI models (Whisper) and ML(TensorFlow.jses) into browserbased applications with minimal effort. An alternative approach that involves open data (ORS/OSM) and the open-source ML may result in a cost-effective solution that can be easily modified by others. The ability to make multiple voice calls at the same time

is facilitated by the client-server architecture, which promotes scalability.

Comparison to Existing Solutions: We are not solely utilizing cloud-based navigation tools like many other presentday navigation aids. The integration of Whisper with the Web Speech API enables users to perform offline actions on Whist being active in browsers. Consequently, the system is economical and reliable in various scenarios.

CONCLUSION AND FUTURE SCOPE

Using OpenAI Whisper, open mapping APIs, and web technologies we have presented an AI-powered voice navigation aid for the visually impaired. With the ability to issue both spoken commands and audio clues, blind users can travel independently and safely with safer navigation options. Additionally, it allows for obstacle alerts and directions. We have implemented a cost-effective, open-stack framework (Python Flask, TensorFlow.js, Web Speech API) that is both reproducible and flexible.

Key Achievements: The design satisfies the requirements of offline-capable voice navigation and real-time environment sensing. It is a backend ASR and frontend object detection tool that merges machine learning to create an all-in-one assistive system. The system aims to achieve Sustainable Development Goals 10 and 11, which include inclusion and mobility for people with disabilities.

Future Work:

Planned enhancements include:

GPS and Transit Integration: Utilizing live public transportation data and turnby-turn GPS for extended travel. Besides

providing access to walk routes, users could also plan trips for buses or trains using voice modes.

Multi-language Support: Diversifying language setups and TTS voices (e.g., Adding new user communities to Hindi, Spanish and vernacular languages is also an objective. The support for multiple languages in Whisper is already present, and incorporating them into the UI will increase its scope.

Dynamic Routing: Utilizing AI to adjust routes on the fly (e.g. avoiding newly reported obstacles or unsafe areas).

Wearable Form Factor: The frontend could be made into a simple wearable or dedicated device, which could make it easier to use than relying on satay. A smart glasses setup that includes a microphone and bone-conducting audio could be utilized as an illustration.

User Testing: Conducting trials with visually impaired users to evaluate usability and gather feedback, quantitatively measuring improvements in navigation accuracy and user confidence.

In conclusion, the proposed voicecontrolled navigation tool highlights the potential of AI and web tools to enhance accessibility. This prototype can be utilized to build on existing assistive platforms that aim to provide universal and equitable mobility for individuals with visual impairments.

REFERENCE

- [1] World Health Organization, "Blindness and vision impairment," WHO Fact Sheet, Aug. 2023. [Online]. Available: <https://www.who.int/news-room/factsheets/detail/blindness-and-visualimpairment>
- [2] S. Nayak and C. B. Chandrakala, "Assistive Mobile Application for Visually Impaired People," *Int'l J. of Interactive Mobile Technologies (iJIM)*, vol. 14, no. 16, pp. 52–59, Sep. 2020
- [3] S. K. Nehal, R. K. Obheroi, A. Anand, and H. Rana, "A Navigation Device with Voice for Visually Impaired People," in *Advances in Intelligent Systems and Computing*, AISC vol. 624, Springer, 2018, pp. 1369–1380
- [4] M. H. Abidi *et al.*, "A comprehensive review of navigation systems for visually impaired individuals," *Heliyon*, vol. 10, no. 11, e31825, May 2024
- [5] C. Valle, X. Areces, and A. García, "Technological Advancements in Human Navigation for the Visually Impaired: A Systematic Review," *Sensors*, vol. 25, no. 7, 2213, Apr. 2025
- [6] OpenAI, "Introducing Whisper," OpenAI (blog), Sept. 21, 2022. [Online]. Available: <https://openai.com/blog/introducingwhisper>
- [7] S. Redij, A. Rao, S. A. Kumar, and A. Jadhav, "Smart Bus Tracking For Blind And Visually Impaired," *ITM Web of Conferences*, ICACC 2022, Apr. 2022
- [8] OpenRouteService, "Directions API," OpenRouteService.org, 2025. [Online]. Available: <https://openrouteservice.org/>
- [9] M. S. A. Baig *et al.*, "AI-based Wearable Vision Assistance System for the Visually Impaired: Integrating Real-Time Object Recognition and Contextual Understanding Using Large VisionLanguage Models," *arXiv preprint arXiv:2412.20059*, Dec. 2024
- [10] J. A. Aderonmu *et al.*, "Assistive

Technology and Sustainable Development Goals,” *Physio-Pedia*, 2023. [Online]. Available:

https://www.physiopedia.com/Assistive_Technology_and_Sustainable_Development_Goals

[11] A. Sharma and B. Singh, “An Intelligent Navigation System for the Visually Impaired Using Speech Recognition and GPS,” *Proc. IEEE Conf.*, 2023. (Details not provided)

[12] R. N. Kandalan and K. Namuduri, “A Comprehensive Survey of Navigation Systems for the Visual Impaired,” *IEEE Trans. Instrum. Meas.*, vol. 68, no. 2, pp. 461–481, Feb. 2019.

[13] D. Gawari and S. Bakuli, “Voice and GPS Based Navigation System For Visually Impaired,” *Int. J. Eng. Res. Applic.*, vol. 2, no. 4, pp. 59–64, 2012.

[14] S. Xu *et al.*, “Wearable Obstacle Avoidance Electronic Travel Aids for Blind and Visually Impaired Individuals: A Systematic Review,” *IEEE Access*, vol. 11, pp. 66587–66613, 2023.

[15] X. Lian, W. Guo, X. Zhang, and Q. Pei, “Outdoor Navigation for Blind People: A Survey,” *Sensors*, vol. 21, no. 3, 957, Jan. 2021.

[16] A. Cordova and M. Tokuda, “Mobility technologies for visually impaired people: a 75-year review,” *Assist. Technol.*, vol. 34, no. 5, pp. 254–267, 2022.

[17] S. Gupta and P. Gupta, “Smart Assistive Navigation System for Visually Impaired People,” *Int. J. Comput. Sci. Inf. Technol. (IJCSIT)*, vol. 11, no. 6, pp. 53–59, 2020.

[18] NVIDIA TensorFlow.js,

“TensorFlow.js Guide,” *TensorFlow.org*, 2023. [Online]. Available:

<https://www.tensorflow.org/js/tutorials> (accessed 2025).

[19] Mozilla Developer Network, “Web Speech API,” MDN Web Docs, 2024.

[Online]. Available: https://developer.mozilla.org/enUS/docs/Web/API/Web_Speech_API (accessed 2025).

[20] M. Dobosz, “Mobile phones as assistive technology,” *Disabil. Rehabil.: Assist. Technol.*, 2025. (In press)

Note: References [11]–[20] include representative related works and general sources. Citation links (e.g. [14†L345L349]) connect to specific content used from each reference.