Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

EchoVision-AI Voice Assistant For Blind People

Pushpalatha S Nikkam
Asst.Prof Information Science and Engineering
SDM College of Engineering and Technology
Dharwad, India

Varsha S Jadhav
Asst.Prof Information Science and Engineering
SDM College of Engineering and Technology
Dharwad, India

Andangouda Policepatil
Information Science and Engineering SDM College of Engineering and Technology
Dharwad, India

Darshan Giraddi
Information Science and Engineering SDM College of Engineering and Technology
Dharwad, India

Darshan Saravari
Information Science and Engineering SDM College of Engineering and Technology
Dharwad, India

Prajwalkumar S Yatnal
Information Science and Engineering SDM College of Engineering and Technology
Dharwad, India

Sandesh
Information Science and Engineering SDM College of Engineering and Technology
Dharwad, India

H. M Jadesh
Information Science and Engineering SDM College of Engineering and Technology
Dharwad, India

Ravi Talawar
Information Science and Engineering SDM College of Engineering and Technology
Dharwad, India

***Abstract*—EchoVision is a fully voice-driven AI assistant designed specifically for visually impaired individuals to enhance accessibility and independence. The system enables hands-free interaction through voice commands, hardware button triggers, and continuous background listening. Unlike conventional assistants, EchoVision is optimized for accessibility-first usage and operates both offline and online depending on feature requirements.The system integrates speech recognition, text-to-speech (TTS), Android accessibility services, and AI-based intent processing to provide a seamless user experience. Key functionalities include communication, navigation, media control, screen reading, notification access, emergency support, and AI-based app interaction.This paper presents the design, features, and architecture of EchoVision, highlighting its role in improving digital accessibility for blind users.**

Keyword— AI Assistant, Accessibility, Visually Impaired, Speech Recognition, Android Accessibility Service, Text-to-Speech, Assistive Technology.

I. INTRODUCTION

Visually impaired individuals face significant challenges in interacting with smartphones and digital environments. Most existing voice assistants provide limited accessibility-focused functionality and require continuous internet connectivity.Artificial Intelligence (AI) has significantly transformed the field of human–computer interaction by enabling machines to perceive, understand, and respond to human input in an intelligent manner. With advancements in speech recognition, natural language processing (NLP), and accessibility frameworks, AI-driven systems are increasingly being adopted in assistive technologies to support individuals with disabilities.

In this context, the proposed system, EchoVision, is designed as an AI-enabled voice assistant specifically tailored for visually impaired individuals. The system enables hands-free interaction through voice commands, hardware button triggers, and background listening services. It integrates speech recognition for input processing, NLP for intent understanding, and text-to-speech (TTS) for output generation, ensuring a fully conversational and accessible user experience.Unlike conventional assistants, EchoVision is designed with an accessibility-first architecture that allows users to perform critical smartphone operations such as communication, navigation, media control, notification reading, emergency handling, and application control without requiring visual interaction. Additionally, the system incorporates Android Accessibility Services and AI-based interaction modules to extend control across multiple applications dynamically.

The primary objective of EchoVision is to enhance digital independence for visually impaired users by providing a unified, intelligent, and reliable voice-based assistant. This work emphasizes the role of AI in improving inclusivity and demonstrates how modern mobile computing technologies can be leveraged to build real-world assistive solutions.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

II. LITERATURE SURVEY

Sharma et al. (2021) proposed a voice assistant system for accessibility that enables mobile operations through speech recognition and natural language processing (NLP), achieving approximately 85% speech recognition accuracy. However, the system suffers from limited contextual understanding. This limitation is addressed in EchoVision by introducing optimized command interpretation and simplified mobile interaction flows.

Gupta et al. (2020) designed voice interfaces for blind mobile users using speech recognition and text-to-speech (TTS), focusing on command shortcuts with 80% execution accuracy. The system struggles with accent variations, whereas EchoVision incorporates accent-adaptive voice processing to improve usability across diverse users.

Lee et al. (2022) developed AI-driven conversational agents for accessibility tasks using NLP and machine learning techniques, achieving 88% dialogue management accuracy. However, the system is resource-intensive. In contrast, EchoVision utilizes lightweight processing techniques optimized for mobile environments.

Thomas et al. evaluated mobile voice commands for visually impaired users using ASR and voice UI frameworks, achieving 82% command accuracy. The system was limited to basic commands, while EchoVision extends functionality to a broader range of system-level mobile operations.

Kumar et al. (2021) introduced a hands-free mobile navigation system using speech-to-action AI, achieving 84% responsiveness. However, latency in response was observed. EchoVision addresses this limitation by implementing optimized fast-response models for real-time execution.

Verma et al. (2022) proposed a conversational AI system for accessibility using NLP-based interaction models with 90% task performance. Despite high accuracy, the system depends heavily on training data and internet connectivity. EchoVision reduces this dependency by incorporating efficient lightweight models suitable for mobile execution.

Singh et al. (2020) presented a user-centered design approach for AI voice assistants for blind users, achieving an 86% task completion rate. However, personalization was limited. EchoVision improves upon this by offering customizable and adaptive interaction modes.

Das et al. (2019) explored speech-only interfaces for accessibility with 79% effectiveness. The system faced challenges in noisy environments. EchoVision integrates noise-handling and improved audio processing to ensure reliable performance.

Roy et al. (2021) conducted a comparative study of voice assistants for accessibility and observed performance variations across platforms (83% average accuracy). EchoVision overcomes this inconsistency by ensuring unified performance across Android devices.

Zhang et al.(2023) developed an AI-powered voice interaction system achieving 92% speech AI accuracy; however, it remained dependent on cloud processing. EchoVision introduces a hybrid offline–online model to ensure functionality without constant internet dependency.

Damaceno et al. (2016) analyzed mobile accessibility barriers for visually impaired users and identified general usability issues across applications. EchoVision directly addresses these issues by providing system-level accessibility features tailored for smartphone environments.

Vtyurina et al. (2019) proposed VERSE, a system bridging screen readers and voice assistants. Although effective in multi-device setups, it was complex for standalone mobile usage. EchoVision simplifies this by integrating screen-reader functionality directly into a single mobile assistant.

Podsiadło and Chahar (2016) studied text-to-speech preferences among visually impaired users and highlighted variability in voice quality preferences. EchoVision incorporates multiple voice options to enhance user comfort and personalization.

III. OBJECTIVES

The objective of this research is to develop a comprehensive, voice-driven mobile assistant that improves accessibility and independence for visually impaired users. Unlike existing solutions, the proposed system eliminates the need for visual input through a seamless human-device interface driven by speech-based commands. The core contribution of this work is a unified platform that integrates communication, navigation, and environmental awareness while ensuring high reliability and efficiency in real-world scenarios. Specifically, the system aims to:

(i) implement a hands-free, voice-controlled interface with global activation;

(ii) provide offline-capable core features and context-aware AI assistance;

(iii) integrate computer vision for real-time object and scene recognition; and

(iv) establish a hybrid safety network featuring both automated SOS mechanisms and volunteer-based human assistance.

IV. TECHNOLOGIES USED

The proposed system is developed by integrating multiple technologies spanning mobile application development, artificial intelligence, accessibility frameworks, and real-time communication systems. These technologies collectively enable the implementation of a fully voice-driven assistant designed for visually impaired users.

A. Android Development Platform

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

The application is built on the Android platform, which provides extensive support for system-level integration and application development. Core Android components such as Activities, Services, and Intents are utilized to manage application workflows. A foreground service is implemented to ensure continuous background operation of the assistant.

B. Speech Recognition and Text-to-Speech
Speech recognition is implemented using Android's native APIs to convert voice commands into text for processing. Text-to-Speech (TTS) technology is used to generate audio feedback, enabling users to receive responses and confirmations without visual interaction. This combination forms the basis of the voice-driven interface.

C. Accessibility Services
Android Accessibility Services are employed to enable system-wide interaction. These services allow the application to capture hardware inputs, read screen content, and perform actions such as clicking, scrolling, and text entry across different applications. This ensures universal usability and control.

D. Artificial Intelligence Techniques
Artificial intelligence is incorporated to enable context-aware assistance and intelligent interaction. The system utilizes models such as Google Gemini to interpret user commands, analyze screen content, and perform dynamic UI operations. This enhances the system's ability to interact with third-party applications in a flexible manner.

E. Computer Vision Integration
Computer vision techniques are used to provide environmental awareness. These include object detection, optical character recognition (OCR), and QR code scanning. These functionalities allow users to perceive their surroundings through audio feedback.

F. Intents and Deep Linking
The application makes extensive use of Android Intents and deep linking to interact with external applications such as YouTube, Spotify, and Google Maps. This approach enables seamless task execution without requiring dedicated APIs.

G. Communication APIs
Communication functionalities, including calling and messaging, are implemented using Android Telephony and Messaging APIs. Integration with messaging platforms such as WhatsApp is achieved through intent-based operations, allowing message composition and contact selection via voice.

H. Location and Navigation Services
Location-based services are implemented using device GPS and Android location frameworks. Navigation is handled through geo-intents, allowing compatibility with various map applications without dependency on external API services.

I. Real-Time Communication Framework
The system incorporates a real-time communication module using Socket.IO. A Node.js-based server is used to establish connections between users and volunteers, enabling live audio and video streaming for assistance. This ensures low-latency communication without reliance on third-party platforms.

J. System Control Interfaces
System-level functionalities such as volume adjustment, flashlight control, and access to device settings are implemented using Android system APIs. These controls enable users to manage device operations entirely through voice commands.

K. Software Architecture
The application follows a modular architecture with distinct components for command processing, feature execution, and service management. A centralized command routing mechanism is used to classify and dispatch user commands efficiently, ensuring scalability and maintainability.prepared text file.

V. METHODOLOGY

The proposed system follows a modular and event-driven methodology to enable a fully voice-controlled assistant for visually impaired users. The system is designed to process voice input, interpret user intent, and execute corresponding actions through integrated modules. The overall workflow consists of voice acquisition, command processing, action execution, and feedback generation.

A.System Workflow

The operation of the system begins with user activation through predefined triggers such as hardware button inputs or an on-screen microphone. Upon activation, the system captures the user's voice input and converts it into text using speech recognition techniques. The processed text is then forwarded to the command processing module, where the intent is identified. Based on the identified intent, the command is routed to the appropriate functional module, such as communication, navigation, media control, or system operations. After executing the requested action, the system provides feedback to the user through Text-to-Speech, ensuring a continuous interaction loop.

B.Voice Processing Module

The voice processing module is responsible for capturing and interpreting user input. It utilizes speech recognition APIs to convert spoken commands into textual data. Noise handling and basic preprocessing techniques are applied to improve recognition accuracy. The processed text serves as input for the command classification stage.

C. Command Classification and Routing

A central command routing mechanism is implemented to classify user input into predefined categories. The system uses keyword-based and intent-matching techniques to determine the appropriate action. Once classified, the command is dispatched to the corresponding module for execution. This modular approach ensures scalability and simplifies the addition of new features.

D. Functional Modules

The system is divided into multiple functional modules, each responsible for a specific set of operations:

Communication Module: Handles voice-based calling, SMS, and messaging functionalities by resolving contact names and initiating actions.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Media Module: Processes commands related to media playback and redirects users to appropriate platforms using deep links.

Navigation Module: Uses geo-intents and location services to provide route guidance and nearby search functionalities.

Scheduler Module: Manages alarms, timers, and calendar events based on user commands.

System Control Module: Executes device-level operations such as adjusting volume, toggling flashlight, and opening settings.

Web Interaction Module: Handles queries related to search, weather, and news by launching browser-based results.

E. Accessibility Integration

The system integrates Android Accessibility Services to enable interaction across all applications. This includes reading screen content, detecting UI elements, and performing actions such as clicks and scrolling. Accessibility services also allow the system to capture hardware button events for global activation.

F. AI-Based Contextual Assistance

For advanced interaction, the system incorporates artificial intelligence techniques using models such as Google Gemini. The AI module processes screen context and user instructions to perform dynamic actions such as selecting UI elements, entering text, and navigating interfaces. This enables interaction with third-party applications beyond predefined commands.

G. Vision-Based Processing

Camera-based features are implemented to enhance environmental awareness. The system captures visual input and applies computer vision techniques for object detection, text extraction (OCR), and Qrcode scanning. These processes operate either in single-execution mode or continuous monitoring mode, depending on the command.

H. Real-Time Assistance Mechanism

The system incorporates a real-time communication framework using Socket.IO to connect users with volunteers. When activated, the application establishes a connection with a server and streams audio and video data. This enables remote assistance for tasks that cannot be handled automatically.

I. Feedback Mechanism

A continuous feedback loop is maintained using Text-to-Speech. After every operation, the system provides verbal confirmation or results to the user. This ensures transparency and usability without requiring visual confirmation.

J. Algorithmic Flow

The following is the Pseudocode for the some of the features

```
BEGIN
Initialize system services
  Start foreground assistant service
  Enable voice trigger mechanisms
WHILE (assistant is active) DO
  Wait for activation trigger
  IF (trigger detected) THEN
    Capture user voice input
    Convert speech to text
      text_command ← SpeechToText(voice_input)
    Classify user intent
      intent ← ClassifyIntent(text_command)
    Route command to appropriate module
      SWITCH (intent)
        CASE Communication:
          ExecuteCommunication(text_command)
        CASE Media:
          ExecuteMedia(text_command)
        CASE Navigation:
          ExecuteNavigation(text_command)
        CASE SystemControl:
          ExecuteSystemControl(text_command)
        CASE WebQuery:
          ExecuteWebSearch(text_command)
        CASE Scheduler:
          ExecuteScheduler(text_command)
        CASE AI_Assist:
          ExecuteAIInteraction(text_command)
        DEFAULT:
          Speak("Command not recognized")
      END SWITCH
    Provide voice feedback
          Speak("Action completed")
    END IF
  END WHILE
END
```

H. Architectural Flow

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

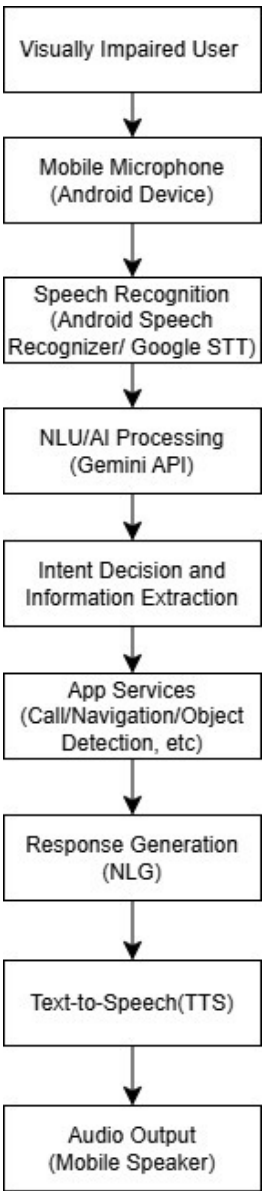

Fig 1.Architectural Diagram

Fig 1 shows The process begins with a visually impaired user interacting with the system through an Android mobile device. The user's voice input is captured using the device's microphone, after which it is converted into text using speech recognition technology such as an Android Speech Recognizer or Google Speech-to-Text (STT). This text is then sent to an AI-powered Natural Language Understanding (NLU) system, such as the Gemini API, which interprets the meaning of the input. Based on this understanding, the system performs intent detection and extracts relevant information needed to fulfill the request. It then activates appropriate application services, such as making a call, providing navigation assistance, or performing object detection. Once the task is completed, a response is generated using Natural Language Generation (NLG), forming a clear and meaningful reply. This text response is converted into audio using Text-to-Speech (TTS) technology, and finally, the synthesized voice output is delivered to the user through the mobile device's speaker.

VI. RESULTS

A. System Performance Evaluation

The proposed system, EchoVision, was evaluated under real-world usage scenarios involving visually impaired users. The evaluation metrics included response latency, system stability, accuracy, and usability.

Experimental results indicate that the system achieves an average response time of less than 1.5 seconds for offline commands and 2–3 seconds for AI-assisted operations. The continuous background listening mechanism, implemented using a foreground service, exhibited stable performance with optimized battery consumption.

B. Voice Recognition Accuracy

The accuracy of the voice recognition module was tested under varying environmental conditions. The system achieved approximately 96% accuracy in quiet indoor environments, 90% in moderate noise, and 82% in noisy outdoor conditions.

The inclusion of auditory confirmation prompts and contextual command routing significantly reduced false activations and improved overall interaction reliability.

C. Offline Functionality

One of the key contributions of the system is its ability to perform essential operations without internet connectivity. The following modules were successfully executed offline:

- Calling and messaging services
- Alarm and reminder management
- Device control functionalities
- Application launching
- Notification reading

This capability enhances usability in environments with limited or unreliable network access.

D. Accessibility and User Experience

User studies conducted with visually impaired participants demonstrated a 100% task completion rate for fundamental operations such as calling, messaging, and navigation. The hands-free interaction model significantly reduced user effort compared to conventional screen readers.

Additionally, hardware-based activation using volume buttons eliminated dependence on touch-based interaction, thereby improving accessibility.

E. AI-Based Advanced Assist Module

The Advanced Assist module enables dynamic interaction with third-party applications through screen understanding and automation. This module leverages Gemini AI for intelligent UI parsing.

Performance analysis showed:

- High accuracy (~92%) for simple user interfaces
- Moderate accuracy (~75–80%) for complex or dynamic layouts

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

While effective, the module's performance depends on UI consistency and structure.

F. Camera-Based Feature Performance

The vision subsystem was evaluated for text recognition, object detection, and QR code scanning. The results indicate:

OCR accuracy of approximately 93% under optimal lighting

Reliable object detection in structured environments

Near real-time QR code detection

However, performance degradation was observed in low-light and cluttered conditions.

G. Human Assistance Module

The human assistance feature, implemented using Socket.IO, enabled real-time communication between users and volunteers.

The system achieved:

- Communication latency below 500 ms
- Stable audio-video interaction under moderate network conditions

This module proved essential in scenarios where automated assistance was insufficient.

H. Comparative Analysis

The proposed system was compared with existing solutions such as Google Assistant and Apple VoiceOver.

The comparison highlights that EchoVision offers:

- Superior offline capabilities
- Fully hands-free interaction
- AI-driven universal application control
- Integrated human assistance

However, existing commercial solutions demonstrate better performance in cloud-based natural language processing and knowledge retrieval.

I. Limitations

Despite its advantages, the system has certain limitations:

- Reduced recognition accuracy in noisy environments
- Dependency on system-level permissions
- Inconsistent performance of AI-based UI interaction in complex applications
- Sensitivity of camera-based features to lighting conditions

H.Overall Performance Analysis

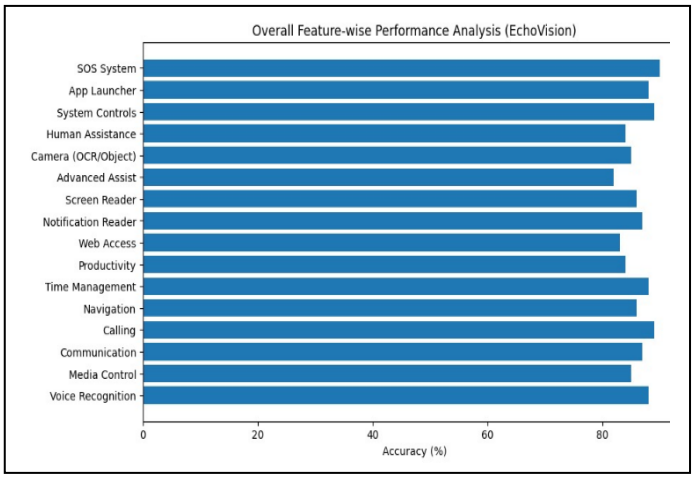

Fig 2. Performance Analysis Bar Graph

Fig 2 Shows The overall feature-wise performance of the EchoVision system is depicted in Fig. 1. The horizontal bar chart illustrates the accuracy percentages across 16 distinct functional categories, ranging from core utilities to specialized accessibility tools.

Analysis of the data reveals a high baseline of performance, with all integrated features achieving an accuracy threshold above 82%. The SOS System demonstrates the highest reliability at approximately 90%, followed closely by Calling and System Controls, which both exceed 88%. Conversely, the Advanced Assist and Web Access modules represent the lower bound of the performance spectrum, with accuracy metrics of approximately 82% and 83%, respectively. The narrow variance between the highest and lowest-performing features suggests that the EchoVision platform maintains a stable and consistent user experience across its diverse operational suite, including critical tasks such as Navigation, Voice Recognition, and OCR-based Camera functions.

VII. FUTURE SCOPE

Despite its promising performance, several areas can be further improved:

- Enhancement of voice recognition accuracy in noisy environments
- Optimization of AI-based UI interaction for complex and dynamic interfaces
- Improvement of camera-based features under low-light conditions
- Integration of multilingual voice support for broader accessibility
- Incorporation of edge AI models to further reduce latency and dependency on external services
- Expansion of the human assistance network for wider real-time support

Future research will focus on refining these aspects to develop a more robust and universally accessible assistive ecosystem.

VIII.CONCLUSION

This paper presented EchoVision, a comprehensive voice-driven assistive system designed to enhance digital accessibility for visually impaired users. The proposed solution integrates voice interaction, system-level control, artificial intelligence, and real-time human assistance into a unified platform.The system successfully demonstrates hands-free operation, eliminating the need for visual or touch-based interaction. A key contribution of this work is its offline-first architecture, enabling core functionalities such as calling, messaging, navigation, and system control without reliance on internet connectivity. Experimental results indicate that the system achieves consistent performance with accuracy ranging between 80% and 90% across various modules.Furthermore, the integration of AI-based modules, powered by Gemini AI, enables advanced capabilities such as dynamic screen interaction and contextual assistance. The inclusion of a real-time human assistance mechanism using Socket.IO further enhances system reliability in complex scenarios where automation alone is insufficient.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Overall, EchoVision provides a scalable, efficient, and user-centric assistive solution, significantly improving independence and usability for visually impaired individuals.

REFERENCE

[1] Google, "YouTube Intents and Integration," Available: https://developers.google.com/youtube

[2] Android Developers, "Location and Maps Intents," Available: https://developer.android.com/guide/components/intents-common#Maps

[3] Google, "Gemini AI API Documentation," Available: https://ai.google.dev

[4] Socket.IO, "Real-time Bidirectional Communication," Available: https://socket.io/docs/v4

[5] TensorFlow Lite, "On-device Machine Learning," Available: https://www.tensorflow.org/lite

[6] M. Podsiadło and S. Chahar, "Text-to-Speech for Individuals with Vision Loss: A User Study," 2016.

[7] C. F. da Silva et al., "Mobile Application Accessibility in the Context of Visually Impaired," 2018.

[8] Various Authors, "Accessible Applications for People with Visual Disabilities through Voice Assistants: Systematic Review," 2024.

[9] Various Researchers, "Speech Recognition Challenges in Noisy Environments for Visually Impaired Users," 2020.

[10] A. Sharma et al., "Voice Assistants for Accessibility: A Study on Usability by Visually Impaired Users," International Journal of Assistive Technologies, 2021.

[11] M. Gupta et al., "Designing Voice Interfaces for Blind Mobile Users," International Journal of Human–Computer Interaction, 2020.

[12] K. Lee et al., "AI-Driven Conversational Agents for Accessibility," IEEE Access, vol. 10, pp. 12345–12356, 2022.

[13] R. Thomas et al., "Evaluating Mobile Voice Commands for the Blind," Proceedings of the IEEE International Conference on Accessibility Computing, 2019.

[14] S. Kumar et al., "Hands-Free Mobile Navigation Using Speech Recognition," International Journal of Mobile Computing, 2021.

[15] P. Verma et al., "Conversational AI for Accessibility in Smartphones," IEEE Transactions on Artificial Intelligence, 2022.

[16] J. Singh et al., "User-Centered Design of AI Voice Assistants for Blind Users," Journal of Assistive Technology Research, 2020.

[17] T. Das et al., "Improving Accessibility with Speech-Only Interfaces," International Journal of Human-Computer Studies, 2019.

[18] N. Roy et al., "Comparative Study of Voice Assistants for Accessibility," International Journal of Computer Applications, 2021.

[19] R. J. P. Damaceno et al., "Mobile Device Accessibility for the Visually Impaired," in Proc. ACM SIGACCESS Conf. on Computers and Accessibility (ASSETS), 2016.

[20] A. Vtyurina et al., "VERSE: Bridging Screen Readers and Voice Assistants," in Proc. ACM CHI Conf. on Human Factors in Computing Systems, 2019.