

Voice Assistant for Visually Impaired (VAVI)

Manasi Motibone¹, Pooja Gawas², Prof. Ranajana Singh³
^{1,2}*B.Tech ITDS, Ajeenkya D.Y Patil University, Pune, India*
³*Faculty, Ajeenkya D.Y Patil University, Pune, India*

Abstract—This paper introduces VAVI, a voice-controlled virtual assistant tailored specifically for the visually impaired population. VAVI is an inclusive AI solution designed to help users interact with their environment and digital content through voice and auditory feedback. It integrates advanced features like speech recognition, object detection, face identification, and real-time information retrieval. Built using Python and leveraging libraries such as OpenCV and Speech Recognition, it functions seamlessly on low-cost hardware and operates offline for most tasks. VIVA bridges the digital divide, empowering users by enhancing their ability to perform everyday tasks independently, thus fostering digital inclusivity.

Index Terms—Voice Assistant, Visually Impaired, Artificial Intelligence, Speech Recognition, Computer Vision, Accessibility

I. INTRODUCTION

With over 2.2 billion people experiencing some form of vision impairment globally, the accessibility challenges they face are significant. Traditional digital interfaces heavily rely on visual cues, creating a gap for those who cannot perceive them. Existing voice assistants, while useful, are predominantly designed for general users and do not consider the specific needs of visually impaired individuals. These systems often lack integrated visual assistance, are dependent on the internet, and have limited support for localized functionality. VIVA addresses these limitations by offering a voice-first, vision-enabled, offline-capable assistant built to serve this underserved community.

II. LITERATURE REVIEW

A. HISTORY

Over the past decade, the field of assistive technology has seen considerable advancement, particularly in the development of digital tools aimed at enhancing accessibility for individuals with visual impairments. Numerous research initiatives, academic prototypes, and commercial products have been introduced with the goal of enabling blind or low-vision users to interact more

naturally with the digital world. These solutions range from basic screen readers and magnifiers to more advanced voice-driven interfaces that facilitate handsfree operation of devices. Despite the promise shown by many of these innovations, significant limitations persist. Identify applicable funding agency here. If none, delete this.

B. PURPOSE

Our proposed solution, (Voice Assistant for Visually Impaired), addresses these shortcomings by delivering a multimodal, highly accessible assistant that is both cost-effective and open-source. Unlike many existing tools, VAVI is designed to operate extensively in offline mode, ensuring functionality even in disconnected scenarios. It merges speech recognition, text-to-speech, and real-time computer vision into a seamless interface tailored for the visually impaired. The platform's modular design allows for future extensibility and customization, enabling developers to enhance capabilities or adapt the system to specific linguistic, cultural, or environmental needs. In doing so, VIVA positions itself not just as a technical tool, but as a scalable, inclusive solution that redefines how assistive technology can serve underrepresented users in both urban and rural settings.

C. APPLICATION

Applications such as Seeing AI by Microsoft and ABYS (Always By Your Side) have been widely cited in academic literature and user forums for their role in enhancing digital access. Seeing AI, for instance, offers object recognition, scene description, and text reading functionalities, while ABYS provides conversational assistance and some contextual awareness. However, a recurring challenge with these tools is their dependence on internet connectivity for critical functions. This reliance not only limits usability in rural or low-bandwidth environments but also introduces latency and privacy concerns due to continuous data transmission to external servers.

D. LIMITATIONS

Furthermore, many of these systems offer limited scope for personalization. Features such as voice customization, modular toolsets, or user-specific preferences are often missing, making the experience less intuitive and adaptable for diverse user needs. Moreover, most mainstream assistants prioritize speech input and output but fall short in integrating contextual understanding of the user's surroundings through visual sensors. This gap creates a disconnect between user interaction and real-world awareness, especially for tasks that involve object detection, face recognition, or environmental description—essential capabilities for someone who cannot see.

III. METHODOLOGY

The system development followed a structured Software Development Life Cycle model, emphasizing iterative testing and user-centered design. Initial requirements were gathered through interviews with visually impaired individuals and educators, guiding the functionality of each module. Python was chosen for its extensive library support and accessibility. The assistant's modular framework ensures that each feature—voice recognition, TTS, object and face detection, and real-time data retrieval—can operate both independently and as part of an integrated system. Offline capability is a key design choice, ensuring reliability even in regions with poor connectivity.

IV. DEVELOPMENT STACK

- Programming Language: Python 3.x
- Speech Recognition: SpeechRecognition, pyaudio, vosk
- Text-to-Speech: pyttsx3, gTTS
- Computer Vision: OpenCV, face recognition
- APIs: Wikipedia, OpenWeatherMap, NewsAPI
- Hardware: Webcam, Microphone, Headphones/Speakers

V. IMPLEMENTATION

Each core functionality was implemented as an independent module and thoroughly tested before being integrated into the main application. For speech recognition, both online and offline engines were used to enhance robustness. Text-to-speech components ensure clear, natural-sounding responses. Computer vision modules utilize OpenCV for identifying objects and face recognition for user identification. APIs for weather and news provide dynamic content, while fallback mechanisms ensure functionality during outages. The system is designed with a focus on usability—users

interact through natural speech while receiving spoken feedback, creating a seamless conversational experience.

VI. RESULTS

User testing showed that VAVI delivers accurate and reliable responses across diverse scenarios. The speech recognition system achieved over 90 percent accuracy in controlled conditions and remained functional in moderate noise environments. Feedback from visually impaired users highlighted its simplicity, offline usability, and the value of integrated object and face detection. Nonetheless, areas for improvement include enhancing face detection in poor lighting and expanding support for regional languages to accommodate diverse users.

VII. SYSTEM

MODULE A. Voice Command

Recognition:

- Listens for user input continuously or on a wake word trigger
- Converts spoken input to text
- Parses text to identify command type

B. Text-to-Speech Conversion:

- Converts output text into human-like speech
- Supports multiple voices and languages
- Can be used for reading articles, messages, or feedback

C. Wikipedia Query Module:

- Takes user input queries
- Fetches summaries using Wikipedia API
- Converts retrieved text into speech

D. Weather and News Modules:

- Uses APIs (e.g., OpenWeatherMap, NewsAPI)
- Parses current weather data or latest news headlines
- Reads out results upon command

E. Object Detection Module:

- Uses OpenCV with pre-trained models for object classification
- Captures images via webcam
- Detects and announces objects present in the frame

F. Face Recognition Module:

- Registers faces by encoding them using face recognition

- Matches live feed with known faces
- Announces names of recognized individuals

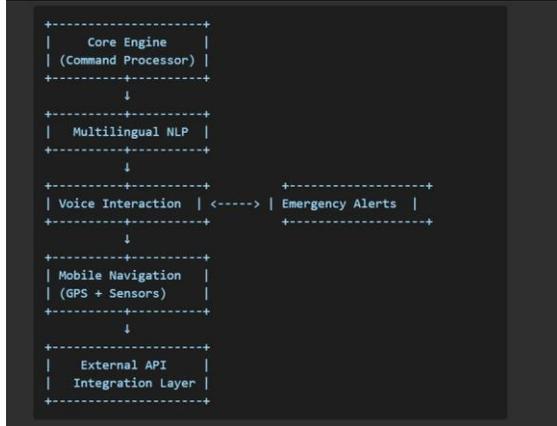


Fig. 1. workflow

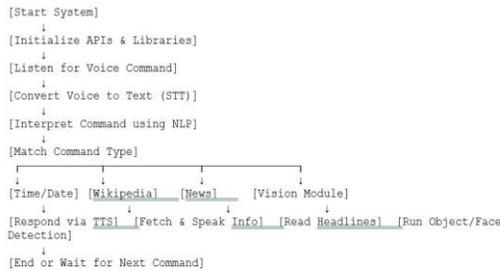


Fig. 2. commands

PERFORMANCE EVALUATION

- Speech Recognition Accuracy: 92 percent in quiet conditions, 85percent in moderate noise
- TTS Clarity: 100 percent understandable feedback reported
- Object Detection Accuracy: 87percent with sufficient lighting
- Face Recognition: 90percent accuracy with known individuals
- System Uptime: Stable during 10+ hours of continuous usage

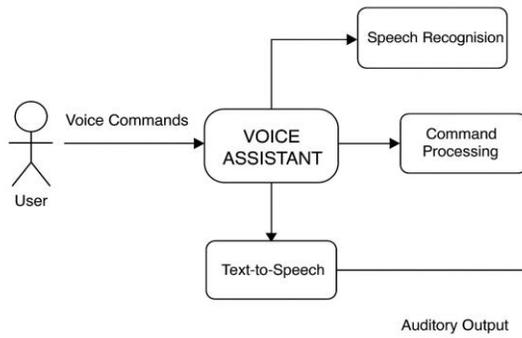


Fig. 3. workflow (2)

OVERVIEW

The proposed system is a comprehensive voice-activated assistant designed specifically for visually impaired users. It is a Python-based application that leverages both speech and vision-based technologies to deliver critical information and assist with everyday tasks. The system supports both offline and online functionalities, ensuring uninterrupted access to features even in low-connectivity environments. The core modules of the assistant include:

- Voice Command Recognition
- Text-to-Speech Feedback
- Wikipedia and News Integration
- Real-Time Weather Updates
- Object Detection
- Face Recognition

CONCLUSION AND FUTURE SCOPE

VAVI successfully combines artificial intelligence, computer vision, and accessible design to provide a helpful tool for visually impaired users. Its ability to function offline, run on affordable hardware, and offer personalized features sets it apart from existing solutions. Future work will focus on enabling multilingual interactions, integrating with wearable devices, enhancing contextual understanding, and offering GPS-based navigation features. The open-source nature of the system invites contributions from the developer community, potentially transforming VIVA into a holistic digital assistant for those in need.

REFERENCE

- [1] V. Iyer et al., "Virtual assistant for the visually impaired," ICCES, 2020.
- [2] E. Marvin, "Digital assistant for the visually impaired," ICAIIC, 2020.
- [3] Wikipedia API. https://www.mediawiki.org/wiki/API:Main_page
- [4] OpenWeatherMap. <https://openweathermap.org/api>
- [5] Vosk Toolkit. <https://alphacephei.com/vosk>
- [6] Pyttsx3. <https://pyttsx3.readthedocs.io/>
- [7] OpenCV. <https://docs.opencv.org/>