Human voice controlled intelligent system assists the disabled people smartly[⋆]

Soumyajit Das[1], Arijit Payne[1], Rishab Sen[1], Ankanendu Mondal[1], Priyash Das[1], and Munshi Yusuf Alam[1][0000−0002−1699−5175]

[1] Budge Budge Institute of Technology, Nischintapur 700138, India
[2] munshiyusufalam@bbit.edu.in
https://www.bbit.edu.in/
[3] {soumyajitdas8420,arijitpayne1,senrishab101,ankanmondal412003,priyashalucard}@gmail.com

Abstract. The increasing adoption of voice-enabled technologies has highlighted their potential to empower individuals with physical disabilities, enabling them to independently navigate digital platforms. However, challenges such as noise interference, limited command vocabulary, and platform incompatibility persist. Our research product, which makes use of state-of-the-art technologies including AI, Natural Language Processing, and Automatic Speech Recognition (ASR), is a Voice-Controlled Integration System. Key concerns including cross-platform usability, noise resilience, and speech recognition accuracy are addressed by the system, which seeks to offer hands-free control over computers and communication platforms for amputees. Through the integration of features like as voice-controlled media playing, speech-to-text, and application navigation, the solution provides increased accessibility and independence with a certain degree of precision.

Keywords: disabled person · voice conferencing · application control · virtual mouse · voice intelligent.

1 Introduction

Voice technologies have a clear future in helping people with physical disabilities specially amputees communicate with the digital world independently. However, a few challenges remained, such as noise interference, limited contextual understanding, and a deficiency in platform compatibility. According to recent studies, around 40% of physically disabled individuals use voice commands for web browsing, yet only 2% of websites are fully assistive technology compatible[4]. Such statistics signify the need for vigorous and universal solutions to accessibility challenges.

There are currently more than 1 billion disabilities in the world, so there is an increasing demand for technologies that can make things accessible and usable.

[⋆] Supported by organization BBIT, Kolkata

[4] https://www.pewresearch.org/publications/

Voice-activated systems based on development in Automatic Speech Recognition and Natural Language Processing can be very promising, but they are often challenged in "real-life" environments, especially in noisy conditions or while supporting multilingual users. Fixing these issues is significant to return the benefits of such systems into the more extensive user community.
Recently, numerous researchers have combined control circuits with intelligent personal assistant (IPA) systems, like Google Assistant, Siri, and Alexa, to take advantage of the natural language processing (NLP) capabilities of these systems. These systems along with the IoT [1] enabled devices have been used extensively to control conventional home appliances [2], computers [3], cars, wheelchairs [4]. Tharaka, L. [5] has suggested a browser that can be accessed by using voice commands to move the mouse pointer and speech to text to enter URLs. Additionally, the mouse cursor may be controlled by computer vision. Both employees with disabilities and regular people can use this web browser to make their jobs easier. Md. Rakibuz Sultan et. al. [6] worked in order to reduce the inconvenience of using a computer and gaining access to household appliances, they conducted a review of the state-of-the-art in assistive technology using voice recognition and developed a voice-controlled desktop application model on the Microsoft Windows platform especially for visually impaired people.
Studies show that mobile communication technology improves independence and social interactions for people with disabilities through better connectivity. Research highlights major obstacles like poor inclusive design and lack of policy support, which could increase economic gaps. Equitable mobile access and reform are needed to ensure technology benefits disabled communities, reducing the digital divide and promoting social inclusion [7]. Our proposal is a voice-controlled integration system that would allow impaired persons to easily write, edit, and format documents for office work and connect with Google Meet to hold meetings efficiently. This system would particularly enhance access, convenience, and independence for users with disabilities. Context-sensitive advanced AI to be integrated into the system improves understanding of users' intentions; it is multilingual, noise-resilient, and capable of offline processing to minimize internet reliance. These features serve to fill the gap that other systems do not cover and at the same time offer priority access and usability.
The important contributions of this work are improvements in context-aware interpretation of voice commands, a privacy-focused-offline functionality, and a modular architecture for customization and scalability. Those gaps that were left open in current technology are now being filled to create the necessary infrastructure for more advanced, user-centric voice software solutions. Subsequent sections of this paper detail system methodology with Data collection, system approach, Experimental setup, the Result section with the implementation challenges, experimental validation, and lastly the broader implications of this work.

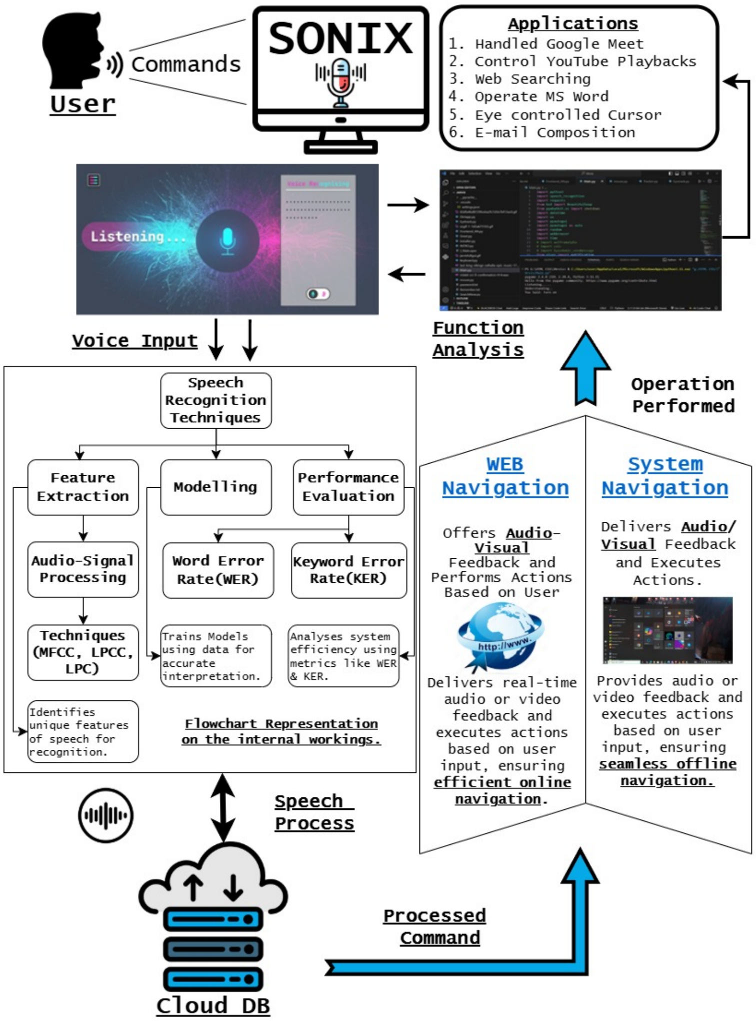

User
Commands
SONIX
Applications
1. Handled Google Meet
2. Control YouTube Playbacks
3. Web Searching
4. Operate MS Word
5. Eye controlled Cursor
6. E-mail Composition
Listening...
Voice Input
Function Analysis
Operation Performed
Speech Recognition Techniques
Feature Extraction
Modelling
Performance Evaluation
Audio-Signal Processing
Word Error Rate(WER)
Keyword Error Rate(KER)
Techniques (MFCC, LPCC, LPC)
Trains Models using data for accurate interpretation.
Analyses system efficiency using metrics like WER & KER.
Identifies unique features of speech for recognition.
Flowchart Representation on the internal workings.
WEB Navigation
Offers Audio-Visual Feedback and Performs Actions Based on User
http://www.
Delivers real-time audio or video feedback and executes actions based on user input, ensuring efficient online navigation.
System Navigation
Delivers Audio/Visual Feedback and Executes Actions.
Provides audio or video feedback and executes actions based on user input, ensuring seamless offline navigation.
Speech Process
Processed Command
Cloud DB

Fig. 1: SONIX: Intelligent Voice Controlled System

2 Methodology

Our project aims to provide a voice-controlled system "SONIX" Figure 1 tailored specifically for individuals with physical disabilities who have lost the ability to use their hands due to congenital conditions or accidents. The methodology is divided into three key phases: Input, Processing, and Output. Each phase has been carefully designed to ensure that the system meets the requirements of these users and provides an accessible and seamless interaction experience.

3 Data Collection and Testing Summary

To better understand user needs and challenges, a usability survey was conducted with eight participants, including individuals with physical disabilities. Participants reported various challenges, such as left-hand paralysis caused by an auto accident, right-hand and right-leg disabilities resulting from a car accident, visual impairments developed due to health conditions, a recent hand fracture from sports, and mobility issues associated with aging. The survey also captured whether these disabilities were congenital or acquired later in life, helping to contextualize their experiences.

When asked about their general computer usage, participants highlighted tasks such as customer data entry, freelancing for editing purposes, learning, official work, and personal activities like social media browsing, YouTube, college assignments, school-related tasks, meetings, and accounting. For learning purposes, participants used computers to engage in activities such as sales-related software training, photo and video editing, researching technology and historical facts, creating presentations, performing data entry, typing, gaming, coding, and working on projects. Some also expressed interest in learning new technologies and financial tools like Tally.

Participants shared the applications and software they commonly use, including Microsoft Office (Word, Excel, etc.), Canva, Adobe, Picsart, Chrome, Edge, social media apps (Facebook, WhatsApp, YouTube), VS Code, and Google Meet. Despite their reliance on technology, frustrations with web accessibility and system navigation were prominent. Challenges included difficulties typing and using both the keyboard and mouse due to disabilities, physical discomfort from sitting for extended periods, trouble reading small text on screens, network issues, problems using the mouse, and the complexity of existing accessibility solutions.

Our research aim is to develop an application prototype to meet all these challenges which have been addressed from our survey. The testing process measured execution times for commands like file opening and application navigation, ensuring fast response times. The system achieved accuracy rates exceeding 90% for most core features across different scenarios. To validate its robustness, the application was tested in noisy environments, demonstrating resilience to background noise. Additionally, the system's multilingual capabilities were evaluated with diverse accents and languages, achieving over 95% accuracy. These results reflect the system's efficiency, user-friendliness, and ability to meet the needs of individuals with physical disabilities. Further refinements are planned based on user feedback to enhance accessibility and usability.

4 System Approach

The development of the Voice-Controlled Intelligent System shown in Figure 1 employs an integrated approach to address critical challenges in accessibility while leveraging advanced technologies. The system design begins with a modular and scalable architecture that ensures seamless integration of hardware and software components. An external microphone is used to capture voice instructions, ensuring high-quality audio signals. Real-time noise cancellation and filtering techniques are applied to reduce interference and create a robust foundation for the system. The back-end, developed in Python, processes captured the speech using Automatic Speech Recognition (ASR) to convert it into text. This is followed by Natural Language Processing (NLP), which interprets the user's intent and maps it to predefined actions. The modular design allows for scalability and customization, making the system adaptable to a wide range of applications.

The technology stack combines efficient tools to enhance the system's performance. The front end is designed using HTML, CSS, JavaScript, ReactJS, and Bootstrap, providing an intuitive and responsive user interface. On the back end, Python is integrated with databases like MongoDB and MySQL for managing user preferences and system logs. Core libraries such as "SpeechRecognition," "PyDub," and "DeepSpeech" enable accurate speech processing, while Text-to-Speech (TTS) engines generate audio feedback. Additionally, the system ensures cross-platform compatibility by seamlessly integrating with applications like Google Meet, Zoom, and Discord, allowing for effective voice navigation and control.

The process flow begins with capturing the user's speech through the external microphone. The audio is processed by the ASR module to transcribe speech into text, even in challenging environments such as noisy conditions or when handling diverse accents. The Python back-end applies NLP techniques to analyze the transcribed text, identify the user's intent, and execute corresponding actions such as opening applications, controlling multimedia playback, or managing documents. The system provides tailored outputs in two formats: audio feedback, delivered through speakers using TTS engines, and visual feedback, displayed as confirmation messages or status updates on the screen. This dual-output capability ensures that the system effectively caters to diverse user needs.

Prototyping focuses on the development of core features, including hands-free document management, real-time voice navigation for conferencing tools, and multilingual support to enhance inclusivity. To ensure its robustness, the system undergoes extensive testing in real-world scenarios, including environments with varying noise levels and accents, as well as offline conditions. Hardware components such as microphones, speakers, and webcams are optimized to ensure minimal latency and high accuracy during real-time processing.

5 Experimental Setup

The three stages of our system architecture Figure 1 are designed to offer an excellent and dependable solution for the aforementioned societal issues. We aim to show in this section how a disabled person will engage with the system and how the system will react to commands. Here, we have considered four amputees named as Person 1, Person 2, Person 3, and Person 4.

5.1 Phase 1: Human voice input

The input phase focuses on how users interact with the system using voice commands. This phase ensures that the system captures voice inputs accurately and processes them effectively. The system is designed to cater to the specific needs of four distinct user profiles which are as follows:

- Person 1: This user needs to manage Google Meet, including opening and closing meetings, enabling/disabling the microphone and camera, and handling other meeting controls.
- Person 2: This user primarily controls YouTube playback, including playing, pausing, adjusting the volume, navigating videos, and enabling full screen mode.
- Person 3: This user performs tasks like browsing Google or Wikipedia, reading aloud search results, and navigating web pages using voice commands.
- Person 4: This user operates Microsoft Word, performing tasks such as opening/closing documents, voice typing, text formatting, and saving files in various formats.

To capture these inputs, the system relies on speech recognition tools that process user voice commands in real-time. The system is designed to ensure compatibility with users who can hear and see properly but cannot physically use traditional input devices like a keyboard or mouse. By collecting and pre-processing the input data effectively, the system ensures that it can cater to the unique needs of each user profile.

5.2 Phase 2: Data Processing

This phase is the core of the system, where voice commands are transformed into actionable tasks. It is built on advanced technologies like Automatic Speech Recognition (ASR) and Natural Language Processing (NLP), supported by robust feature extraction techniques.

Speech Recognition and Feature Extraction - Speech recognition begins with the conversion of spoken words into text. To enhance recognition accuracy, feature extraction methods such as Mel-Frequency Cepstral Coefficients (MFCC), Linear Predictive Cepstral Coefficients (LPCC), and Linear Predictive Coding (LPC) are employed. These techniques identify unique features of the audio signals, making it easier to interpret user commands accurately.

Handling Environmental Noise - The system is designed to work effectively in both noisy and noise-free environments. Noise-resilience testing ensures that commands are recognized accurately, even in challenging settings, by leveraging noise reduction algorithms.

Model Training and Performance Evaluation - The system is trained using diverse datasets to ensure high accuracy in interpreting voice commands. It's efficiency is measured using performance metrics like Word Error Rate (WER) and Keyword Error Rate (KER), ensuring that the model consistently delivers reliable results.

Command Processing - Based on the user's profile and needs which are mentioned above, the system processes commands to execute specific tasks:

- Person 1: Voice commands are processed to automate meeting controls in Google Meet, such as starting meetings and toggling the mic or camera.
- Person 2: Commands are interpreted to manage YouTube, including playback control, navigation, and volume adjustments.
- Person 3: Search-related commands are processed to perform Google and Wikipedia searches, display search results, and read them aloud.
- Person 4: Commands are processed to manage Microsoft Word, including document creation, voice typing, text formatting, and file-saving operations.

For some commands, the system fetches data from a cloud-based database. This ensures that users receive accurate and updated results in real time, especially for tasks requiring internet connectivity.

5.3 Phase 3: Output based on functionalities

The output phase of the system generates results based on the processed inputs. The system provides tailored feedback and executes actions categorized as web-based navigation (requiring an internet connection) or system-based navigation (offline operations). Both categories deliver audio-visual feedback and perform the necessary actions, ensuring efficient task completion and user satisfaction based on different functionalities.

Functionality for Each User:-

Person 1: Managing Google Meet For users who need to manage G-Meet, the system leverages web-based navigation to create and control meetings. This includes opening a browser, creating and joining the meeting, and managing controls like enabling/disabling the microphone and camera through voice commands. Functions are shown to execute those commands like When the user says, "Start a new meeting," the system opens a browser, navigates to G-Meet, and automatically creates a meeting link while offering audio and visual feedback.
Functions Used:

- pyautogui/auto: Automates mouse and keyboard actions for navigating G-Meet.
- speech_recognition: Processes voice commands to control meeting actions.
- webbrowser: Opens Google Meet links automatically.
- os: Launches the browser using system commands.
- time: Adds delays for smoother automation.
- PyQt5: Provides a graphical interface for navigation.
- selenium.webdriver: Automates browser actions such as joining meetings and muting/unmuting.
- smtplib: Sends email invitations for meetings.

Person 2: Managing YouTube and Multimedia Integration For users controlling YouTube, the system supports tasks such as playing videos, pausing,

adjusting volume, enabling full screen, and navigating playback. It can also play stored or previously downloaded audio or video content offline. Depending on the task, it uses both web-based and system-based navigation.
Functions Used:

- pywhatkit: Plays YouTube videos via voice commands. speech_recognition: Processes voice commands for video selection and playback.
- webbrowser: Opens YouTube links directly.
- requests & bs4 (BeautifulSoup): Scrapes and fetches video/audio download links from the internet.
- selenium.webdriver: Automates video downloading and playback processes.
- pyautogui: Controls browser actions such as scrolling and playback.
- pygame.mixer: Plays downloaded audio files offline.
- notifypy.Notify / plyer.notification: Sends notifications when downloads are completed.
- random: Randomly selects videos from playlists.
- os: Manages downloaded files and media.

Person 3: Web Browsing and searching For users focused on browsing, the system performs tasks like Google or Wikipedia searches, fetching and reading content aloud, and navigating through search results.
Functions Used:

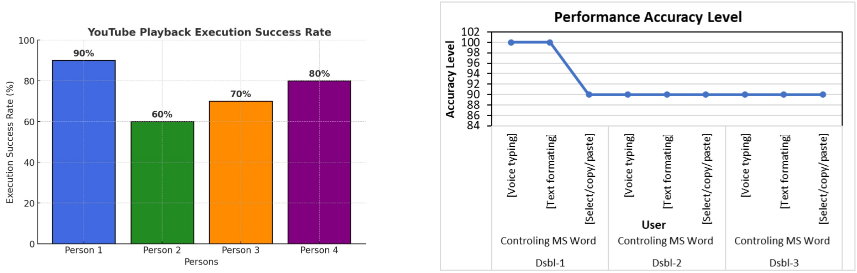
- requests & bs4 (BeautifulSoup): Scrapes and fetches search results or content from Wikipedia and Google.
- pywhatkit: Executes quick web searches directly using voice commands.
- speech_recognition: Processes voice-based queries like "Tell me about Avengers."
- webbrowser: Opens search results in the browser automatically.
- wolframalpha: Provides computational and detailed answers to complex queries.
- selenium.webdriver: Automates search processes and page navigation.
- pyautogui: Automates navigation actions like scrolling or clicking links.
- time: Adds delays for smooth browsing.
- PyQt5: Builds a user interface to assist with browsing tasks.

Person 4: Managing Microsoft Word Documents For users working on documents, the system handles tasks such as opening Microsoft Word, voice typing, formatting text, and saving files in various formats. These tasks rely on system-based navigation, ensuring functionality without internet dependency.
Functions Used:

- lib.Word: Interacts with MSWord documents for editing and formatting.
- pyautogui: Automates typing, selecting, and formatting actions.
- os: Opens, saves, and manages files locally.
- time: Adds delays for smoother automation.
- speech_recognition: Enables voice commands for document operations.
- pywhatkit.sc.shutdown: Closes Word after completing tasks.
- PyQt5: Provides a user-friendly interface for Word file management.
- smtplib: Sends documents via email, if required.
- plyer.notification: Sends notifications when tasks are completed.

6 Results and Discussion

Four participants were used to assess the system's performance referred in Figure 2a, focusing on task execution success rates in real-time scenarios. Persons 2 and 4 had success rates of 80%, Person 1 had 90%, and Person 3 had 70%. The average success rate for all participants was 80%, showing the system's dependability. Tasks like YouTube playback were included, with execution speed and command accuracy as important performance indicators. In this result section, out of four persons, we have described the performance accuracy of three persons named as Disable 1(Dsbl-1), Disable 2(Dsbl-2), and Disable 3(Dsbl-3). The figure 2b depicts the performance of three impaired users while controlling Microsoft Word tasks. When users try to type in MSWord, style the text, or pick a piece of the text to edit, Dsbl-1 achieves 100% accuracy, whilst Dsbl-2 and Dsbl-3 reach 90% accuracy on average. Three disabled individuals, Dsbl-1, Dsbl-2, and Dsbl-3, attempted to meet in a Google Meet. As the meeting host, build a meeting connection using voice and then send it via WhatsApp. The three users conduct operations such as requesting to join, using the camera/mic, chatting, minimizing or maximizing the window, and exiting. Dsbl-1 achieves more than 90% proper execution accuracy across all tasks. However, Dsbl-2 and Dsbl-3 completed the majority of the right uses with a less rigorous deviation has been shown in figure 3a. The following figure 3b depicts the precision of performance of three impaired users (Dsbl-1, Dsbl-2, and Dsbl-3) when manipulating a virtual mouse by gesturing. It includes tasks like eye cursor movement, hand cursor movement, and click-works. Eye cursor movement has an accuracy of 90-100%, whereas hand cursor movement has an accuracy of 85-95%. Click-works have the lowest and inconsistent accuracy among users.

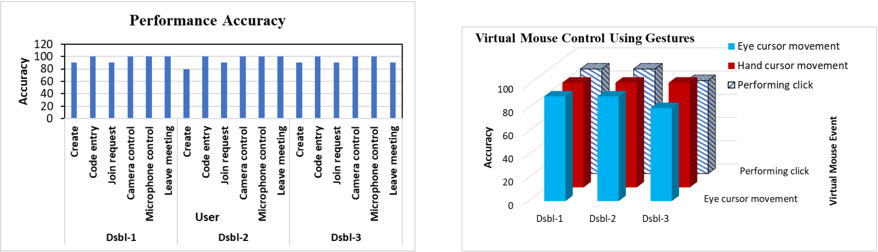


(a) Task Completion Performance (b) MS Word controlled by Disabled person

Fig. 2: Performance level of different applications

Conclusion & Future Scope

The application runs on the Microsoft Windows operating system and is tested by four disabled individuals with different accents and voices. An eye gesture



(a) Video Conference (Google Meet) Performance

(b) Virtual Mouse Controlling performance using gestures

Fig. 3: Accuracy in Microphone controlling & virtual mouse controlling during Google meet through voice control

virtual mouse allows the user to write in Microsoft Word with formatting capabilities, manage and conduct video meetings, and operate installed applications. In addition to sending and receiving emails, this model allows you to open any mail browser, check your junk folder, clean out your inbox, and browse the internet without any issues. Also, the model might eventually be able to zip any file, copy it, and transfer it across the system as well as platform independent with suitable and reliable system security.

References

1. Ali, Abd-elmegeid Amin, et al. "Development of an intelligent personal assistant system based on IoT for people with disabilities."Sustainability 15.6 (2023): 5166.
2. Mtshali, Progress, and Freedom Khubisa. "A smart home appliance control system for physically disabled people."2019 Conference on Information Communications Technology and Society (ICTAS). IEEE, 2019.
3. Isyanto, Haris, Ajib Setyo Arifin, and Muhammad Suryanegara. "Design and implementation of IoT-based smart home voice commands for disabled people using Google Assistant."2020 International Conference on Smart Technology and Applications (ICoSTA). IEEE, 2020.
4. Puviarasi, R., Mritha Ramalingam, and Elanchezhian Chinnavan. "Self Assistive Technology for Disabled People-Voice Controlled Wheel Chair and Home Automation System."IAES Int. Journal of Robotics and Automation 3.1 (2014): 30.
5. Tharaka, L. D. S., D. U. Vidanagama, and W. M. S. R. B. Wijayarathne. "Voice Command and Face Motion based Activated Web Browser for Differently Abled People." (2022).
6. Sultan, Md Rakibuz, and Md Moinul Hoque. "Abys (always by your side): A virtual assistant for visually impaired persons." 2019 22nd International Conference on Computer and Information Technology (ICCIT). IEEE, 2019.
7. Alper, Meryl. Giving voice: Mobile communication, disability, and inequality. MIT Press, 2017