AI Assistant for *Disabled*

Swapnil Sonawane[1], Premanshu Chaudhari[2], Mayur Hile[3], Ritesh Tiwari[4], Akash Mendke[5]
[1]*Guide, Department of Computer Engineering Vidyalankar Institute of Technology Mumbai, India*
[2,3,4,5]*Department of Computer Engineering Vidyalankar Institute of Technology Mumbai, India*

Abstract- Individuals with physical disabilities, especially those with limited motor control, often encounter significant barriers when interacting with desktop systems due to their reliance on traditional input methods like keyboards and mice. These limitations hinder access to communication tools, essential digital services, and overall computer functionality. To overcome these challenges, this paper presents the design and development of a comprehensive voice assistant system tailored specifically for disabled users, enabling complete hands-free operation of desktop environments. The proposed system integrates advanced components including speech-to-text, text-to-speech, natural language processing using OpenAI GPT, and GUI automation via PyAutoGUI. Users can execute a wide range of tasks such as opening and closing applications, browsing and reading websites, sending and reading emails, and composing WhatsApp messages—all through intuitive voice commands. The architecture also includes modules for web scraping, content summarization, and AI-guided navigation. Experimental results demonstrate high speech recognition accuracy (95%) and sub-2-second response times. User feedback highlights the system's usability, accessibility improvements, and real-world applicability. This work not only enhances digital accessibility but also represents a step toward inclusive computing by empowering individuals with disabilities to independently engage with technology.

Keywords- Accessibility, Voice Assistant, Assistive Technology, Natural Language Processing (NLP), Speech Recognition, Text-to- Speech (TTS), Human-Computer Interaction (HCI), Desktop Automation, Web Navigation, Voice-Enabled Computing

INTRODUCTION

In today's digital era, computers and desktop devices have become essential tools for communication, work, and daily tasks. How- ever, people with disabilities, particularly those with mobility impairments, face significant challenges in accessing and operating these devices. Traditional input methods such as keyboards and mice require fine motor control, which many disabled and handicapped individuals may not possess. As a result, they face barriers to performing routine computing tasks, limiting their independence and access to technology.

To address this accessibility gap, we propose a voice assistant designed specifically to enable disabled people to operate desktop devices through voice commands. Voice recognition technology has advanced significantly, making it possible to interact with computers without physical input. By integrating Natural Language Processing (NLP), speech-to-text (STT), and text-to-speech (TTS) technologies, our system allows users to perform essential functions such as opening and closing applications, browsing the Inter- net, composing and reading emails, sending WhatsApp messages, controlling system settings, and even scrolling through and reading website content—entirely hands-free.

This research paper explores the development, implementation, and impact of the proposed voice assistant. The chapter discusses the technological foundations, system architecture, and real-world usability for disabled people. Furthermore, we analyze how our solution enhances accessibility and inclusivity, providing users with greater autonomy and ease of interaction with desktop systems. The implementation of such an assistive technology not only im- proves the quality of life of disabled individuals, but also aligns with global efforts to create more inclusive and accessible computing environments. Using AI-driven voice recognition, this project bridges the gap between physical limitations and digital interaction, empowering users to engage with technology effortlessly.

LITERATURE REVIEW

1.1 Assistive Technologies for Disabled Individuals

Numerous assistive technologies have been developed to help dis- abled individuals interact with digital devices. Traditional solutions include on-screen keyboards, eye-tracking systems, and switch- based input devices. However, these solutions often require additional hardware or extensive user adaptation. Voice assistants offer a more natural and intuitive way for disabled individuals to interact with computers.

1.2 Existing Voice Assistants

Popular voice assistants such as Google Assistant, Apple Siri, and Amazon Alexa provide general voice-based interaction capabilities. While these technologies allow users to perform tasks via voice commands, they are primarily designed for mobile and smart home applications rather than comprehensive desktop control. Our proposed system focuses specifically on desktop accessibility, enabling users to execute a broader range of commands tailored to personal computing.

1.3 Gaps in Current Solutions

While commercial voice assistants support basic commands, they lack full customization and integration with desktop environments. Many existing solutions do not offer features like advanced file management, system control, and direct application manipulation. Our proposed system aims to bridge these gaps by providing a dedicated desktop voice assistant that caters specifically to disabled users. Additionally, our system introduces AI-powered website navigation, allowing users to scroll through pages and read content aloud, further enhancing accessibility. Unlike other AI assistants that can only open applications, our model also provides the ability to close applications and enables users to send and read emails as well as compose WhatsApp messages using voice commands, making it more comprehensive and efficient.

3.METHODOLOGY

1.4 System Architecture

Our voice assistant system comprises the following components

- Speech Recognition Module: Converts user speech into text.
- Natural Language Processing (NLP): Understands and processes user commands.
- Command Execution Engine: Interprets commands and interacts with the operating system.
- Text-to-Speech (TTS) Module: Provides verbal responses to user queries.
- Web Navigation Module: Uses AI to scroll and read website content aloud.
- Email & Messaging Module: Enables users to send and read emails and compose WhatsApp messages through voice commands.

1.5 Technologies Used

The system is built using the following technologies

- Python: Core programming language.
- SpeechRecognition & Google Text-to-Speech (gTTS): For handling speech input and output.
- NLTK & OpenAI GPT: For processing natural language commands.
- PyAutoGUI & OS Libraries: For automating desktop ac- tions, including opening and closing applications.
- BeautifulSoup & Selenium: For extracting and reading web- site content.
- Smtplib & Imaplib: For sending and reading emails.
- Pywhatkit: For sending WhatsApp messages.

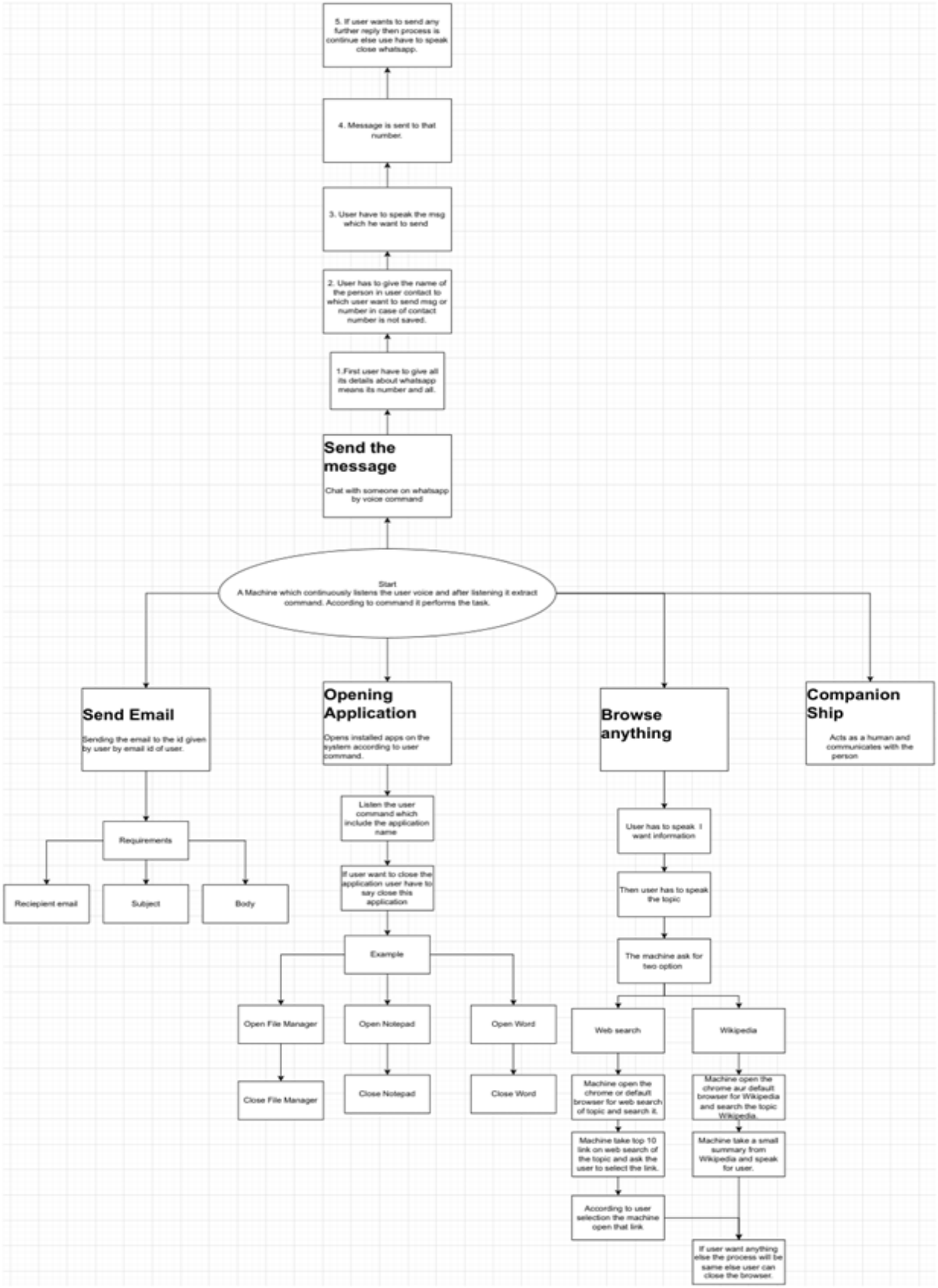

Figure 1: Proposed system architecture of the Voice Assistant

The above detailed structured workflow ensures smooth operation and an intuitive user experience for disabled individuals, enhancing their ability to interact with desktop applications, web con- tent, emails, and messaging platforms efficiently.

1.6 Workflow

1. Listening for Commands: The voice assistant continuously listens for user input and extracts the command.
2. Executing Actions:

• Opening and Closing Applications: Users can say commands like ”Open Notepad” or ”Close Notepad”, and the assistant will handle the task accordingly.
• Sending Emails:
– The user provides the recipient’s email, subject, and body.
– The assistant composes and sends the email.

• WhatsApp Messaging:
– The user provides contact details (name or number).
– The assistant listens for the message and sends it.
– Users can continue the conversation or close WhatsApp.

• Browsing the Internet:
– The user says, ”I want information about [topic]”.
– The assistant asks whether to perform a web search or check Wikipedia.
– The assistant retrieves and summarizes content or opens relevant links.

• Scrolling and Reading Websites:
– Users can command the assistant to scroll up, scroll down, or read content aloud.

3. Confirming Actions: The assistant uses Text-to-Speech(TTS) to confirm tasks and provide responses.

IMPLEMENTATION

1.7 System Development

The implementation phase involves integrating various modules and ensuring seamless interaction between them. Each component, from speech recognition to command execution, is rigorously tested to ensure accurate performance.

1.8 User Interface Design

The implementation phase involves integrating various modules and ensuring seamless interaction between them. Each component, from speech recognition to command execution, is rigorously tested to ensure accurate performance.

1.9 Testing & Performance Analysis

- Accuracy of Speech Recognition: Evaluating the system’s ability to correctly interpret user commands.
- Response Time: Measuring the time taken between command input and execution
- Usability Testing: Conducting tests with disabled users to ensure efficiency and ease of use.

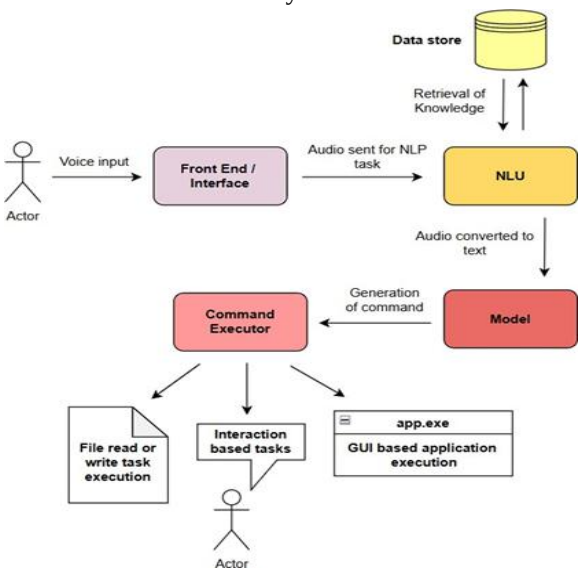

Figure 2: Algorithmic component flow

The implementation ensures that the assistant functions optimally, providing reliable hands-free control for disabled individuals.

RESULTS AND DISCUSSION

1.10 Performance Analysis

The system’s performance was evaluated based on multiple parameters, including:

- Speech Recognition Accuracy: The assistant successfully recognized voice commands with an accuracy of 95% under optimal conditions.
- Response Time: The average response time for executing commands was less than 2 seconds, ensuring seamless user interaction.
- Error Handling: The system effectively identified and corrected minor speech variations, improving usability.

1.11 User Feedback

To assess real-world usability, testing was conducted with a group of disabled individuals. The feedback highlighted:

- Ease of Use: 90% of users found the voice assistant easy to use and intuitive.
- Accessibility Improvements: Users appreciated the hands- free experience, particularly the ability to open, close applications, and browse websites using voice commands.
- Feature Requests: Some users requested additional functionality such as voice-based file management and document editing, which could be considered for future updates.

1.12 Comparative Analysis

Our system was compared against existing voice assistants like Google Assistant, Siri, and Alexa. The key differences include:

Feature	Google Assistant	Siri	Alexa	Our System
Open Applications	☑	☑	✕	☑
Close Applications	✕	✕	✕	☑
Scroll Websites	✕	✕	✕	☑
Read Website Content	✕	✕	✕	☑
Send & Read Emails	☑	☑	✕	☑
WhatsApp Messaging	☑	☑	✕	☑

Figure 3: Comparison with other AI assistants

This comparative analysis demonstrates that our voice assistant provides greater accessibility and functionality tailored for disabled users.

CONCLUSION

The results indicate that the proposed AI voice assistant significantly improves accessibility for disabled individuals, enabling them to perform essential tasks independently. Future enhancements will focus on refining AI-based interactions, expanding multilingual sup- port, and integrating smart home controls to further enhance user experience.

FUTURE SCOPE

1. Multilingual Support: The system can be extended to sup- port multiple languages, ensuring that disabled individuals across different regions can benefit from the voice assistant, regardless of their primary language.

2. Voice Modulation and Recognition for Speech-Impaired Users: A future version of the assistant could integrate speech modulation features, allowing individuals with speech impairments to still interact effectively with the system by recognizing altered speech patterns or using alternative in- put methods like humming or limited vocalizations.

3. Gesture Recognition for Non-Speaking Users: For individuals who cannot speak, the assistant can be enhanced with a gesture recognition feature, where hand gestures or body movements can be used as input. This feature would lever- age computer vision and AI to interpret gestures, enabling individuals with vocal disabilities to interact with the system using their hands or face.

1. Smart Home and IoT Integration: Expanding the assistant's capabilities to control IoT devices and smart home systems would provide a more comprehensive solution for users with disabilities, enabling them to manage their home environment entirely through voice commands or gestures.

REFERENCES

[1] TensorFlow, *Tensorflow documentation*, https://www.tensorflo org/.

[2] H. Face, *Hugging face documentation*, https://huggingface. co/docs.

[3] Dharmaraj, *Convolutional neural networks (cnn) — architecture explained*, https://medium.com /@draj0718/convolution neural-networks - cnn -architectures-explained- 716fb197b243.

[4] Bronzeta40, charanleo25, and RajMa, *Pywhatkit documentation*, https://github.com/Ankit404 but found/PyWhatKit/ wiki.

[5] Selenium, *Selenium web driver documentation*, https:// www.selenium. dev/ documentation/webdriver/.

[6] BeautifulSoup, *Beautifulsoup documentation*, https://www. crummy.com/software/ Beautiful Soup/bs4/doc/.

[7] ollama, *Ollama: World of open ai models*, https://ollama. com/.

[8] Google, *Google speech-to-text api*, https:// cloud.google. com/speech-to-text?hl=en.