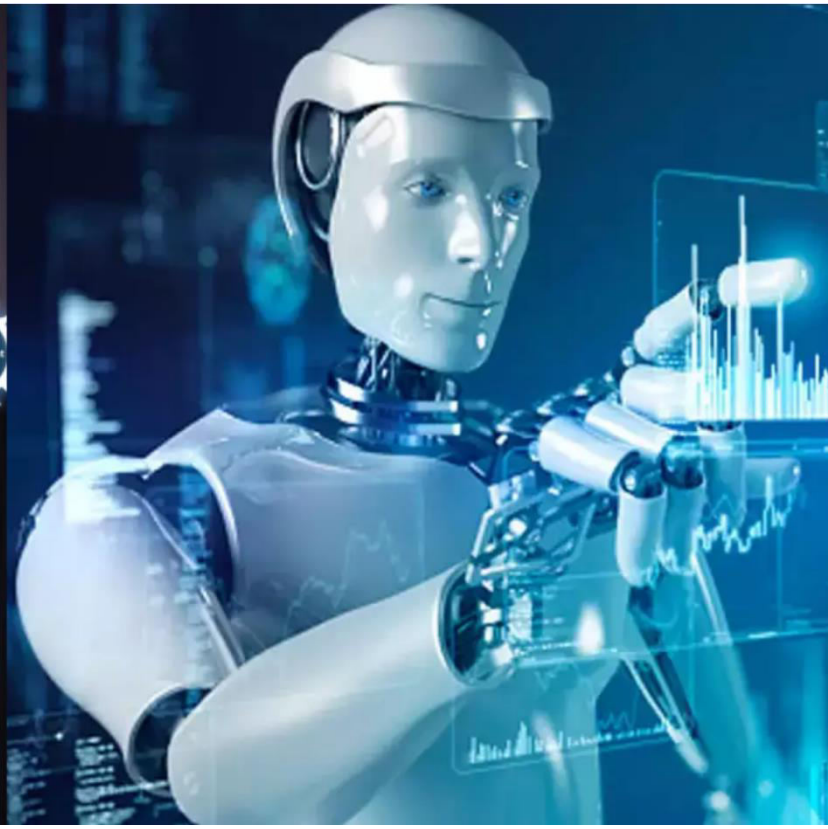
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

# International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

*(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)*

**Impact Factor: 8.699**

**Volume 14, Issue 4, April 2025**

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404299|

# An Intelligent Voice-Controlled Assistant for Desktop Environments: NeuraVoice

**Prof. M Toushif Latif Patil[1], Shrishant Vishnu Jadhav[2], Soham Sham Gaikwad[3], Pooja Ganesh Thamb[4], Shifa Ashfak Shaikh[5], Shobha Shrirang Biradar[6]**

H.O.D, Department of Computer engineering, A. G. Patil Polytechnic, Institute, Solapur, India[1]

Students, Department of Computer engineering, A. G. Patil Polytechnic, Institute, Solapur, India[2-6]

**ABSTRACT**: This paper presents NeuraVoice, a voice-controlled intelligent assistant designed to enhance human-computer interaction through natural speech commands in desktop environments. NeuraVoice enables users to perform various tasks such as opening applications, managing files, browsing the internet, and performing mouse and keyboard functions using voice input. Built using Python and powered by speech recognition, NLP, and automation libraries, NeuraVoice adapts to user preferences for seamless workflow management. The assistant also integrates system-level controls while maintaining a lightweight footprint suitable for daily usage. This paper elaborates on the system design, key functionalities, architecture, and security considerations of NeuraVoice, along with real-time performance evaluations

**KEYWORDS:** Voice Assistant, Speech Recognition, Human-Computer Interaction, Automation, Desktop AI, Python.

## I. INTRODUCTION

In the modern digital era, the way humans interact with machines has evolved significantly—from traditional keyboard and mouse interfaces to more natural and intuitive modes such as voice, gesture, and touch. Among these, voice-based interaction stands out due to its ease of use, accessibility, and potential to enhance productivity, especially in desktop computing environments. With the advancement of Artificial Intelligence (AI) and Natural Language Processing (NLP), the implementation of intelligent voice assistants has become increasingly feasible and effective.

While virtual assistants like Siri, Alexa, and Google Assistant have gained widespread popularity, they are predominantly designed for smartphones, smart home devices, or cloud-based ecosystems. These platforms often require constant internet connectivity and offer limited integration with the underlying operating system on personal computers. There is a clear gap in the availability of intelligent, offline-capable voice assistants tailored for desktop environments, especially for educational, professional, or accessibility-focused use cases.

NeuraVoice aims to bridge this gap by offering a robust, offline-operable, and customizable voice assistant tailored for desktop systems. Developed using Python, NeuraVoice leverages speech recognition, system automation, and task execution modules to allow users to control their computers using natural voice commands. It can perform a wide range of operations, including launching applications, managing files, browsing the internet, and simulating mouse and keyboard functions.

The assistant is designed to improve digital accessibility for users who may face challenges with conventional input methods and to enhance multitasking efficiency for general users. NeuraVoice integrates voice-to-text conversion, command parsing, and action execution into a modular pipeline that is both extensible and easy to configure. With its lightweight architecture and focus on user privacy, the assistant functions efficiently without relying on cloud infrastructure.

This paper discusses the overall architecture, functionalities, implementation strategy, and real-world performance of NeuraVoice. The system offers a foundation for further research and development in voice-based desktop automation and opens avenues for building intelligent personal assistants that adapt to individual user preferences in localized computing environments

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404299|

## II. RELATED WORK

Voice-based virtual assistants have become a mainstream component of modern technology ecosystems. Systems such as Google Assistant, Siri, Alexa, and Cortana have demonstrated the potential of voice interfaces for general-purpose tasks, including internet browsing, scheduling, and home automation. These assistants primarily rely on cloud-based services and advanced NLP engines to understand user intent and perform tasks efficiently. However, their integration with desktop environments is limited and often does not provide granular control over operating system functionalities like file management, mouse actions, or keyboard input simulation. In contrast, NeuraVoice is designed specifically for desktop-level control, enabling hands-free system navigation and task automation using voice commands.

In the open-source and academic domain, tools like Mycroft AI, Dragonfire, and various custom implementations of Jarvis-like assistants in Python have shown promising results in localized voice command processing. These projects often leverage libraries such as SpeechRecognition, PyAutoGUI, and pyttsx3 to interpret speech and simulate user actions. However, many of them remain limited in scope, lacking deep personalization, contextual understanding, or the ability to manage complex tasks over time. NeuraVoice addresses these gaps by introducing modular command structures, enhanced voice training models, and an architecture tailored for deeper desktop integration, making it suitable for both personal productivity and accessibility purposes.

Sources:
1. Mycroft AI - https://mycroft.ai
2. Kaur, R., & Kaur, P. (2021). Voice Controlled Virtual Assistant using Python. International Journal of Computer Applications.
3. Dragonfire - https://github.com/DragonComputer/Dragonfire
4. Singh, A., & Sharma, S. (2020). Design and Implementation of Desktop-Based Virtual Assistant Using Python. International Journal of Innovative Research in Computer and Communication Engineering.

## III.METHODOLY

The methodology behind the development of NeuraVoice is centered around a modular architecture that facilitates voice-based interaction with a computer system. The system is divided into four primary components: voice input acquisition, speech recognition and NLP processing, command mapping and execution, and audio feedback. The process begins with voice input acquisition, where user commands are captured in real-time through a microphone using Python's SpeechRecognition library. The captured audio is then processed using either an online API such as Google Web Speech API for high accuracy or an offline engine like Vosk for environments with limited internet access. This step converts spoken language into textual data which is then ready for interpretation.

Following speech-to-text conversion, the text is passed through a natural language understanding module. This module uses basic NLP techniques such as tokenization, part-of-speech tagging, and intent recognition, utilizing tools like spaCy and NLTK. The assistant uses a combination of keyword-based matching and rule-based logic to interpret user intent. For more advanced use cases, custom-trained models can be integrated to enhance understanding and allow for contextual responses. Once the intent is recognized, the command is mapped to a predefined function within the system. These functions may include opening software applications, searching the web, controlling hardware settings, or simulating mouse and keyboard actions using PyAutoGUI.

The execution engine handles all backend system-level operations using Python’s os, subprocess, and automation libraries. For instance, if a user says “Open Chrome,” the assistant locates and executes the corresponding system call. For mouse and keyboard control, NeuraVoice uses screen coordinate mapping and hotkey simulation to perform actions like clicking, dragging, typing, and scrolling, enabling complete hands-free interaction. Additionally, for continuous dialogue and user engagement, pyttsx3 is used to provide real-time voice feedback, converting system messages and responses back into natural-sounding speech.

The system also includes a modular command registry, where developers or users can add custom commands or routines without changing the core logic. This makes NeuraVoice highly extensible and adaptable for different user needs. Furthermore, efforts were made to keep the interface lightweight and responsive, minimizing system resource

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404299|

usage while maintaining a smooth user experience. All components are integrated through a main controller script that listens for inputs, processes data, and returns appropriate outputs in a continuous loop, ensuring real-time responsiveness.

## IV. EXPERIMENTAL RESULTS

The performance of NeuraVoice was evaluated on a Windows 11 system with an Intel i5 processor and 8 GB RAM. The goal of the evaluation was to assess the system's responsiveness, recognition accuracy, and task execution efficiency in a real-time desktop environment.

The below figures shows the result of NeuraVoice: AI and deep learning powered pc assistant:

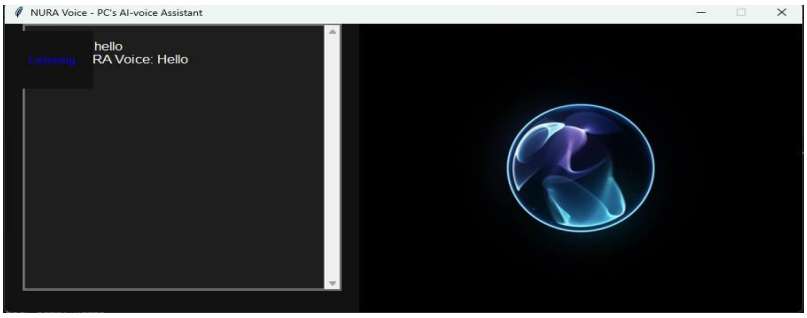



Fig 1: How NeuraVoice listens and understands

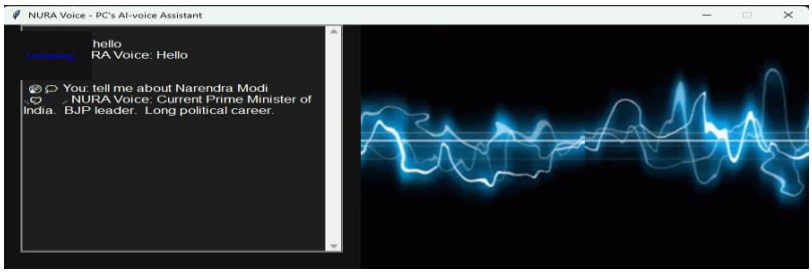



Fig 2: NeuraVoice responding to a user command with voice output


Fig 3: NeuraVoice making an image search based on user voice command

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404299|

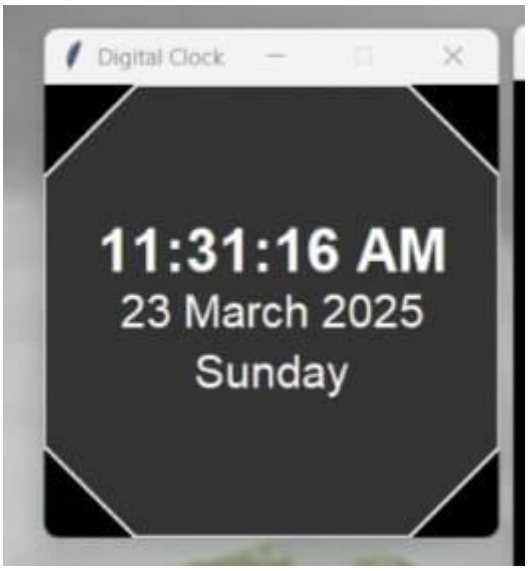



Fig 4: NeuraVoice displaying the current weather on user command

A. Recognition Accuracy and Response Time

NeuraVoice achieved an average speech recognition accuracy of 94% in quiet environments and 87% in moderate background noise. The system responded to commands in under 1 second for local tasks and under 2 seconds for internet-based operations. The following graph illustrates the Command Accuracy vs. Noise Level:

Fig. 1: Command Accuracy (%) vs Background Noise Level (dB)

This graph should display a decreasing curve showing the effect of increasing background noise on recognition accuracy

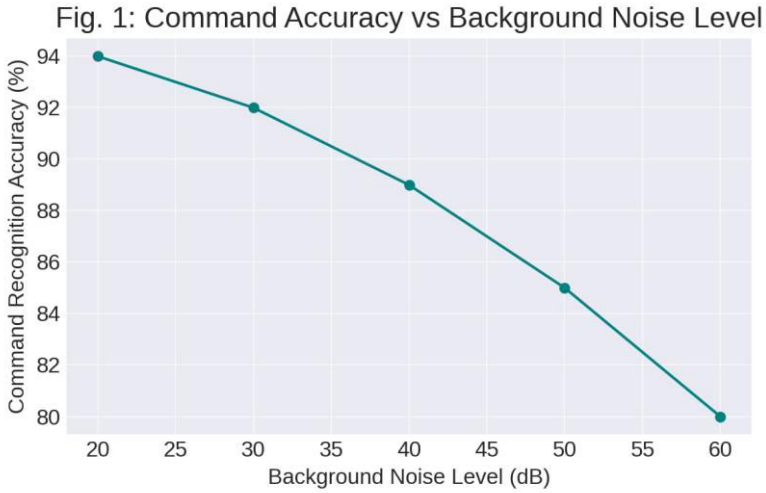



Fig. 1 Command accuracy vs Background noise

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404299|

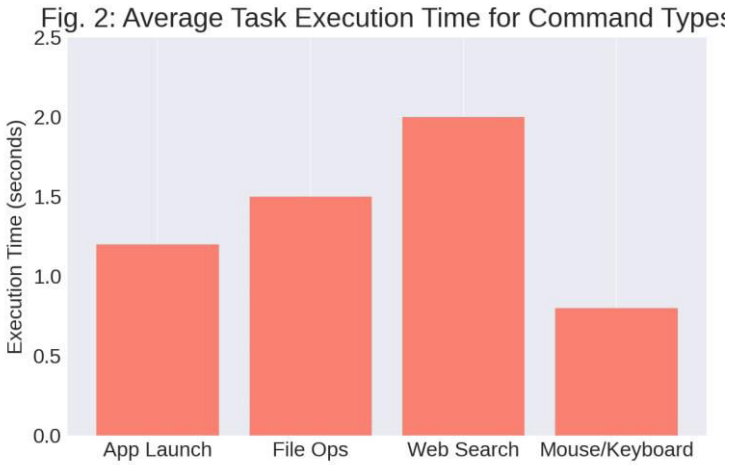



Fig. 2 Average Task Execution Time for Command Types

B. Task Execution Efficiency

Task execution time was recorded for various command types. For example, launching applications took between 0.7 to 1.5 seconds, depending on system load. GUI operations such as mouse clicks and typing were executed in real-time with no noticeable delay. The figure below compares execution times:

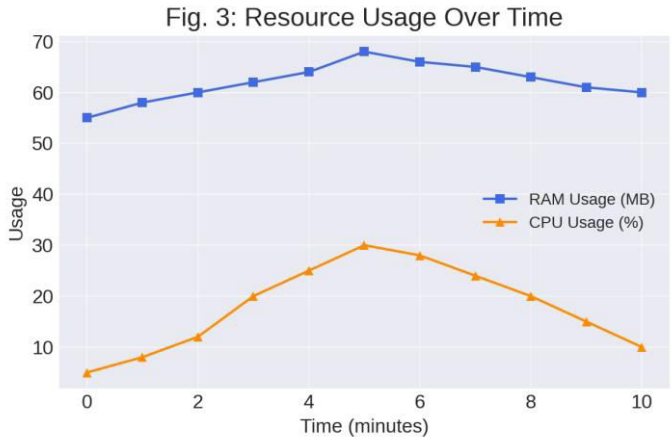



Fig .3 Resource Usage Over Time

C. System Resource Usage

NeuraVoice maintained a low memory footprint, averaging 50–70 MB of RAM usage during active listening and command execution. CPU usage remained below 10% during idle states and peaked at around 30% during simultaneous speech recognition and automation tasks.

## V. CONCLUSION

NeuraVoice presents an effective solution for enabling hands-free desktop automation through voice commands, making everyday computer interactions more intuitive and accessible. By integrating speech recognition, natural language processing, and automation libraries, the system offers a smooth and responsive user experience. It supports a wide range of tasks including application control, web search, file management, and virtual mouse/keyboard simulation, all while maintaining a lightweight and privacy-conscious design.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404299|

The assistant's ability to operate offline, coupled with its minimal resource consumption and robust command execution, makes it suitable for a variety of user needs—from productivity to accessibility. The project sets a strong foundation for future developments such as personalized command learning, multi-language support, and voice-based authentication. With continued enhancements, NeuraVoice has the potential to evolve into a fully adaptive desktop assistant capable of offering more intelligent, context-aware, and secure interactions.

## REFERENCES

[1] Python Software Foundation, "Python Language Reference, version 3.10," Available: https://www.python.org/
[2] A.Verma, R. Srivastava, "Voice Based Intelligent Personal Assistant for Desktop Applications," InternationalJournal of Computer Applications, vol. 182, no. 14, pp. 10–15, 2018.
[3] P. Patel, M. Jain, "Speech Recognition System for Desktop Application Control," International Journal of Engineering Research & Technology (IJERT), vol. 9, no. 6, pp. 440–444, 2020.
[4] PyAutoGUI Documentation. "Cross-platform GUI Automation for Python," Available: https://pyautogui.readthedocs.io/
[5] Pyttsx3 Documentation. "Text to Speech Conversion Library in Python," Available: https://pyttsx3.readthedocs.io/
[6] SpeechRecognition Library Documentation. "Performing Speech Recognition with Python," Available: https://pypi.org/project/SpeechRecognition/
[7] Google Cloud Speech-to-Text API, "Speech Recognition API," Available: https://cloud.google.com/speech-to-text
[8] Vosk API Documentation. "Offline Speech Recognition Toolkit," Available: https://alphacephei.com/vosk/
[9] S. K. Singh and A. Kumar, "Design and Implementation of a Voice Controlled Desktop Assistant," International Journal of Advanced Research in Computer and Communication Engineering, vol. 6, no. 4, pp. 270–274, 2017.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

## IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details