iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 Issue: VI Month of publication: June 2023

DOI: https://doi.org/10.22214/ijraset.2023.54437

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VI Jun 2023- Available at www.ijraset.com

System Volume and Brightness Control Using Voice Commands

Kiruthick K[1], Arin B[2], Yerramsetty Sai Naga Sabarish[3], Dr. Swarnalatha P[4]
School of Computer Science andEngineering, Vellore Instituite of Technology, Vellore, TamilNadu, India

Abstract: This paper proposes a missing feature of windows voice assistant Cortana that voice assistants of android Operating Systems, iPhone Operating Systems. Android operating system has Google Assistant and IOS has Siri, which can adjust the system volume and brightness followed by user'svoice commands, while Cortana can't adjust the system volumeand brightness.
Keywords: Voice Recognition, Human-ComputerInteraction, Voice Interaction, Volume Control, BrightnessControl.

I. INTRODUCTION

Voice communication is a convenient and accessible wayfor users to interact with computers. A voice assistant is an application, which listens to the user's voice and respond to their voice command whatever they instructed the system to do, but a voice assistant can't do whatever the user giving as command, every voice assistant will have some restrictions. In android and IOS the voice assistants Siri and Google Assistant, can listen to the user's voice command and respondaccordingly, especially they can adjust the brightness and volume of the system with respect to user's voice commands.In windows also there is an inbuilt voice assistant called "Cortana", unlike other voice assistants like Siri and Google Assistant, it can't adjust the system volume and brightness. Ifwe ask Cortana to adjust with system volume or brightness, itwill simply say "I'm sorry, but I can't help with that.". So, wehave proposed a windows application, which can adjust system volume and brightness by user's voice commands.

In our proposed prototype, Human-Computer Interaction will be done through voice communication, as user will give the input via voice and system will acknowledge the user about the output with voice. Our system will listen to the user's voice command and start responding to the voice command with adjustment in system volume or brightness when the user starts the services by saying 'start' or 'start services' to their computer through microphone. Why starting services because, this is a voice interaction prototype and needs to listen while the program is running and while running if the user has speaking to someone else the system will recognize and respond to their command. So if the user don't want to use the service but they want the program to run in background for future usage, they can pause the program and start again whenever they want.

Possible commands for adjustments can be stored and matched while user give their commands. After starting service, if the user gives a voice command, then the voice command will be matched with the desired action, and the desired action will be carried out. If none of the commands match in the dataset, then the similar commands to the given voice command will be fetched and showed and outputted bythe system speaker to the user. The user can also pause the service by saying a command and can start again after a while. The user is also allowed to terminate the program using voice command, by saying voice commands like "exit", and a confirmation will be done with the user whether they want toterminate the program and after it will terminate the program.

II. METHODOLOGY

In Our proposed prototype, we are going to use python for controlling system sound and brightness, recognizing user'svoice to text and voice output.

A. *Voice Recognition*

Using the microphone connected to the computer, user's voice commands will recognized and converted into text using python's "speech-recognition" library. It is a library providedby google. The voice will be recorded for 3 seconds, and almost matching text to the voice will be chosen as text and output from this module.

B. *Command Matching from Command Dataset*

The text output from voice recognition module will be matched to the already created voice commands in the Command Dataset. Commands are separately created for operations such as Increase volume or brightness, decrease volume or brightness, mute or unmute volume, half the volume, half the brightness, setting maximum brightness or volume, setting minimum volume or brightness. Most possible commands are stored in a python file as python lists.If the voice command from the user does not match the commands specified in the created dataset, then almost all matching commands will be fetched and output to the user.

C. Volume Control Module
If the user's command is for adjusting volume like increase, decrease, mute, unmute or setting to maximumlevel, this module will be called. Using "pycaw", "ctypes", "comtypes" python libraries, system volume is getting adjusted. System sound will be measured in "decibel", if the system volume is 0 decibel, then the system volume is at its maximum level, if the system volume is -65.25db (for the tested computer) then the volume is 0 (minimum level). Thisminimum level may vary for each computer, so we must findthe minimum decibel volume and compute the adjustments accordingly. According to the user's command the system volume decibel is set, and the system volume will change accordingly.

D. Brightness Control Module
If the user's command is for adjusting brightness like increase, decrease, setting to maximum level, this module will be called. Using "screen_brightness_control" pythonlibrary, screen brightness is controlled. Screen brightness willrange from 0 to 100 percent, so according to the user's command the brightness level will set. If the user is instructed to increase the brightness, it will get increased by step-by- step increment.

E. Voice Output Module
For every action carried because of user's voice command will be output to the user via voice output via system speaker and the same text will be displayed on the screen. Instructions for the users will also be spoken via speaker. Using "pyttsx3" python library text input is converted into voice output.

III. REQUIREMENTS SPECIFICATION

A. Software Specifications
Windows operating system which supports Python3 and Python libraries which include Pyaudio, screen_brightness_control, Pyttsx3, speech_recognition, Ctypes, Comtypes, Pycaw.

B. Hardware Specifications
Microphone, 2 GB Ram minimum, 10 GB Hard diskspace Minimum, Speaker, Active Internet.

IV. HIERARCHICAL TASK ANALYSIS

GOAL: Use voice commands to adjust system volume andbrightness.

A. Steps involved in the HTA
1) Starting the application to make the microphone tolisten for voice commands.
2) Then the user will give their voice commands.
3) Acknowledgement for their voice commands willbe received through voice output.
4) If the user command matches volume adjustment, volume will get adjusted, else if the user commandmatches to brightness adjustment, brightness will get adjusted.
5) If the user command does not match with thecommands stored, a similar command will beacknowledged to the user.
6) Termination of program after user says exit.

B. Diagramatic representation of the HTA

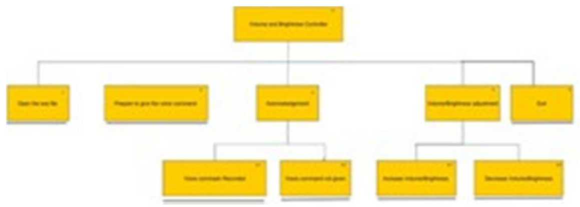


Figure 1: diagrammatic representation of HTA

V. STATE TRANSITION NETWORK

A. Diagramatic representation of STN

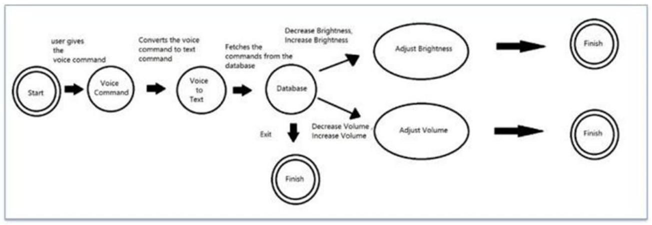

user gives the voice command
Converts the voice command to text command
Fetches the commands from the database
Decrease Brightness, Increase Brightness
Start
Voice Command
Voice to Text
Database
Adjust Brightness
Finish
Exit
Decrease Volume, Increase Volume
Adjust Volume
Finish
Finish

Figure 2: diagrammatic representation of STN

The user can terminate the program whenever they want,that is the reason behind three finish states in STN.

VI. LITERATURE SURVEY

A. Volume Control using Gesture [1]

This paper proposed a volume controller using hand gestures, which uses hand gestures as input to control the system, OpenCV module is basically used in this implementation to control the gestures. This the system consists of a high-resolution camera to recognize the gesturetaken as input by the user. This type of hand gesture system provides a natural and innovative modern way of non-verbalcommunication.

B. Automatic Brightness control of the handhelddevice display with low illumination [2]

The paper addresses the problem of how to adjust the brightness of handheld device displays under low illumination conditions. The authors propose a new algorithm to automatically adjust the brightness of the display based on the ambient light level, the content displayed on the screen, and the user's visual acuity using front camera present in the smartphone. Overall, the paper provides a solution to the problem of adjusting the brightness of handheld device displays under low illumination conditions, which has implications for improving user comfort and productivity.

C. Interaction with Gaze, Gesture, and Speech in a Flexibly Configurable Augmented Reality System [3]

This paper focuses on the design and development of an interactive augmented reality (AR) system that allows users to interact with virtual objects using gaze, gesture, and speech. The system is designed to be flexible, allowing users to customize the interaction methods to suit their preferencesand requirements. They also discuss the user interface and how the system handles the input from the user, particularly with respect to gaze, gesture, and speech. Overall, the paper demonstrates the potential of interactive AR systems that allow users to interact with virtual objects using multiple input modalities, and the importance of designing such systems to be flexible and customizable to meet the needs ofdifferent users.

D. Speech-to-text recognition for multilingual spoken data in language documentation [4]

The paper discusses the use of speech-to-text recognition technology for transcribing spoken data in endangered languages, which can help with languagedocumentation and preservation efforts. The authors focus onthe challenge of multilingual spoken data, which often involves code-switching and other linguistic complexities. The results of the experiment showed that the speech-to-text recognition technology was effective in transcribing multilingual spoken data, with an overall accuracy rate of 76%. Overall, the paper demonstrates the potential of speech-to-text recognition technology for language documentation and preservation efforts, particularly in the context of multilingual spoken data.

E. Free-hand gestures for music playback: derivinggestures with a user-centred process [5]

The paper presents a user-centered approach to deriving free-hand gestures for controlling music playback on mobiledevices. The authors conducted a user study to collect and analyze gesture preferences and designed a gesture set basedon the results. They then evaluated the proposed gestures in asecond user study and found that many participants found thegesture set easy to learn and use. The paper concludes that user-centered design can help create intuitive and effective gesture interfaces for music playback on mobile devices.

VII. STAKEHOLDERS IDENTIFICATION

1) *Users:* People who will use our program foradjusting their system volume and brightness.
2) *Developers:* The group of people who build andkeep the program running.

VIII. TESTING

A. Usability Testing

As our system is based on VUI (Voice User Interface), the test for usability is different from others. Here are someUsability Testing's for Voice Interaction platforms:

1) Involve The Target Audience In The Development From TheBeginning

The targeted users for using our proposed systemcan be peoples who watch movies or videos for long time,people who reads or watch videos while eating or doing other works simultaneously, so that they do not need to involve their hands for adjusting the system volume or brightness.

2) Define the Test Tasks Precisely.

Tasks are defined precisely as Adjusting system volume, adjusting system brightness, Knowing the currentbrightness, Knowing the current volume level.

3) Choose your Testers and Questions Well

Speech commands are chosen accordingly withquestioning the targeted users.

4) We should never expect users to do what we want them to do.

Users don't need to memorize the commands, if they forgot about the exact command, they could give a Similar command and the system will suggest them the similar commands.

B. User Acceptance Testing

S.No	Test description	ExpectedOutcome	Observation	Result
1	Whether the Voice Recognition, recognizes voice to text properly	Exact matchof voice to text	In the presence of lesser noise, inputted voice came as text output.	Pass
2	Whether the voice commandinput obtains the desiredoutput.	Adjustment of volume or brightness with respectto voicecommand	Adjustment done successfully with respect tovoice commands.	Pass
3	Whether feedback received for wrong commands	If commandis not found, acknowledgment is needed for the user.	Acknowledgement received and similar commands outputted through voice.	Pass
4	Whether that pause and startservices working fine.	If a programis paused, it should not doany actions ifit receivescommands until it startsagain.	After programpaused, it listens to thevoice command andnot respondinguntil it receivesstart	Pass
5	Termination of program confirms the user,whether they wantto terminate the program.	After 'exit'command received, it should askconfirmation until it receives 'no'or 'yes' commands.	After 'exit'command received, the system asksconfirmation after confirmation, it terminates.	Pass

C. Testing of Response Time

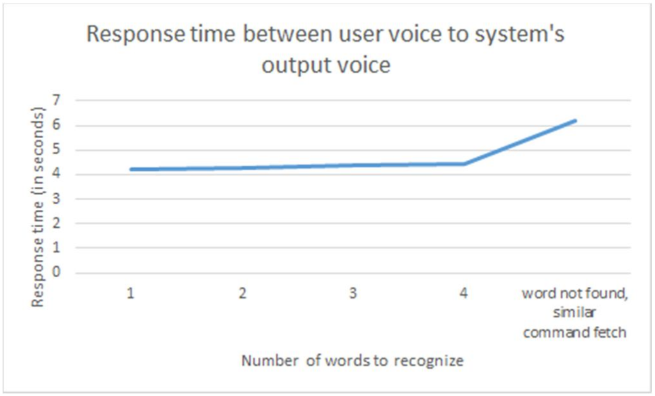


Figure 3: graphical notation of response time analysis

IX. APPLY OF NORMAN'S PRINCIPLES

1) *Visibility:* This program that shows how much time is left until a recording is complete. And shows the text which got converted from the voice command given by the user, So thatuser can confirm whether their desired action is done correctly, also there is voice output to inform the users that their query has been successfully executed.
2) *Feedback:* This is a sound that plays when the voice command is given and voice after the command gets matched with the database confirming that the action was successful or unsuccessful. If the voice command of the user is mismatch to the commands, there is also a voice output as a response, if the Command matches the database commands, then the system will voice output the similar commands.
3) *Constraints:* Until the user uses a voice " start " command the program won't start. Same with the commands after the program starts the user has to give voice commands to adjustthe brightness and volume. If the user don't want to use the system for a while they can say "stop" or "pause" to pause theprogram for a while, where it recognize the text but won't doany actions. If the user say "start" again it will start again. Ifthe user wants to terminate the program he can say "exit", after confirming the user the program will terminate.
4) *Mapping:* This program lets the user know after the action such as adjustment of the volume or brightness. The user willhave proper mapping of the control and the effects. And if theuser says command to mute the volume, it will let the user know that system volume is going to get muted, because aftermute, the voice output can't be heard by the user.
5) *Consistency:* Since this is a voice interaction program, it doesnot need any interface, but for user's acknowledgement, usercan see the input output in the terminal, in which the python file is running. This program runs in a terminal which is beingused many users overall global and which is universally same. The user won't be having any issue using the program. User's input will displayed from the right side of the screen and therepl y (the output) will be displayed in the left side of the screen, as like chat box's to achieve consistency.
6) *Affordance:* The program gives the user voice and text instructions on what must be done and what will happen after those controls.
7) *Mental Model:* The program is programmed in a way which the user can give his/her comfortable voice commands and the program matches it with relevant commands from the database, so the user won't have memorized any new commands to adjust his/her brightness and volume of their system. If they say any relevant commands that do not match from the database of commands, it will say the matching commands, so that user can use that to fulfil the desired action.

X. ANALYSIS

A. KLM

User gives a command for adjusting volume/brightness.

1) Open the setup.exe file from the desktop
2) User gives the voice command
3) User gives exit command

a) Open the setup.exe file from the desktopRemarksOperations
Moves hand to mouse H
Point on the setup.exe file on the desktop PDouble Click B B (left click)

b) User gives the voice command
RemarksOperations
Think and gives voice command "Start" to start the programM
Think and gives command for either to adjust thevolume/brightness M
Wait for operation to be executed M

c) User gives exit commandRemarksOperations
Thinks and gives "Exit" command M
Total time: (H P B B) + (M M M) + (M)
: (0.12 +1.10+0.20+0.20) +(1.35+1.35+1.35) +
(1.35)
: 7.02 sec

XI. ACKNOWLEDGMENT

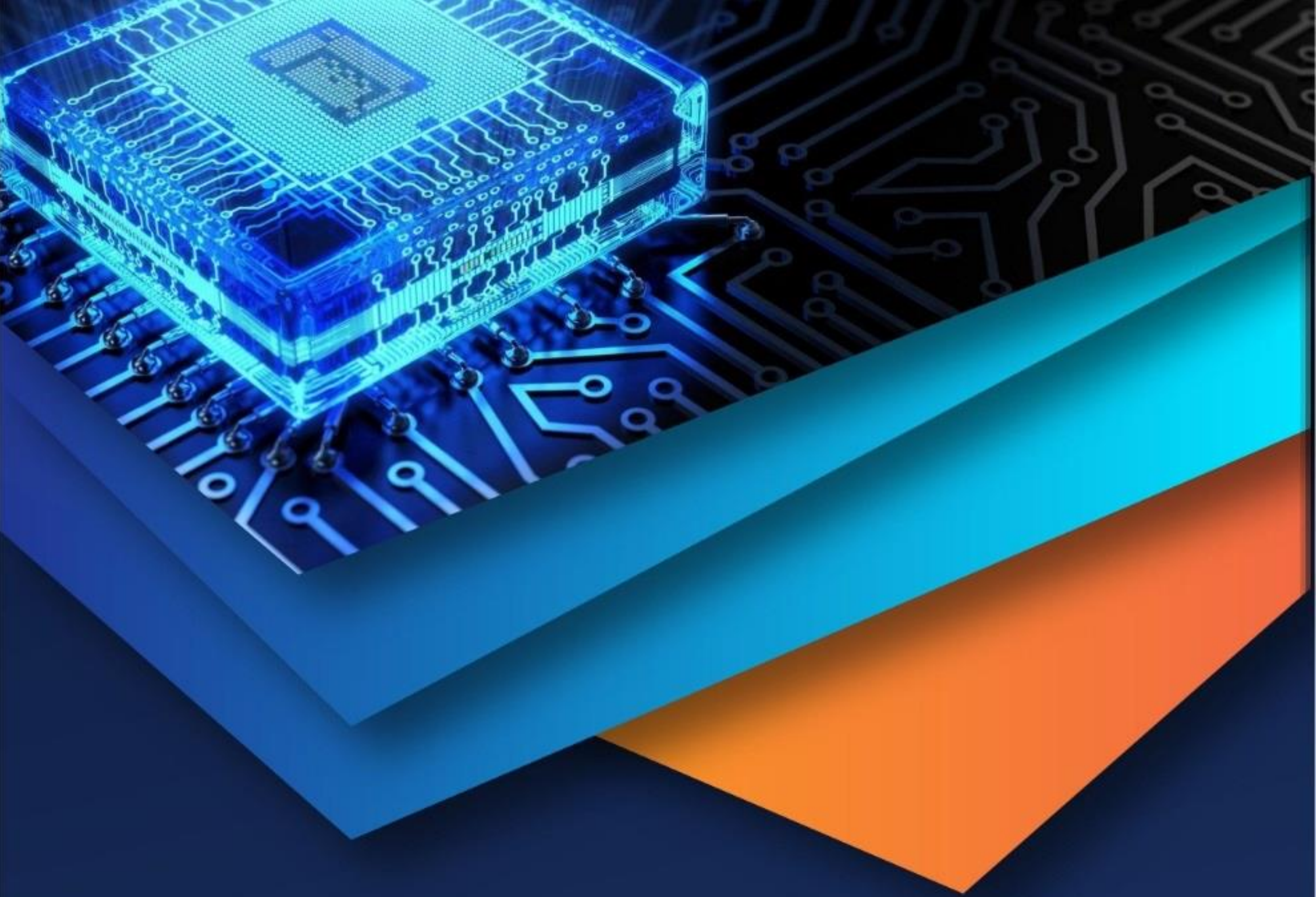
We would like to extend our sincere appreciation toeveryone who helped get this project done. This job would not have been possible without their help. The following is acknowledged by us: Thank you to Dr. Swarnalatha P, our academic advisor, for your essential advice, knowledge, and ongoing assistance throughout the study process. The volunteers who contributed their time and ideas to the usability testing, without whom it would not have been feasible to conduct this study. Vellore Institute of Technologyfor making data and tools available that are necessary for carrying out the research. Friends who helped with programming, data analysis, and other technical facets of theproject. Our research and our knowledge of the subject have been affected by earlier scholars and authors. The ethical andregulatory organizations that checked for compliance with rules and laws and approved our research. For their constant support, inspiration, and motivation throughout the study process, we thank our family and friends. Each team memberbrought their own special skill to the project as they worked together and contributed to various areas of it. Finally, we would like to extend our gratitude to all the authors and scholars whose work has served as a great source of information and inspiration for us. We deeply appreciate all ofthese people and groups for their essential contributions to thisinitiative and are appreciative of them all.

XII. CONCLUSION

As we have implemented speech recognition with Google's API, we can only record the speech and then recognize it. Tomake it effective, we need to implement real time speech recognition, which requires paid API but makes Human- Computer Interaction much faster and better.

REFERENCES

[1] Singh, M. P., Poswal, A., & Yadav, E. Volume Control using Gestures.
[2] T. -Y. Ma, C. -Y. Lin, S. -W. Hsu, C. -W. Hu and T. -W. Hou, "Automatic brightness control of the handheld device display with low illumination," 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE), Zhangjiajie, China, 2012, pp. 382-385, doi: 10.1109/CSAE.2012.6272797.
[3] Z. Wang, H. Wang, H. Yu and F. Lu, "Interaction With Gaze, Gesture, and Speech in a Flexibly Configurable Augmented Reality System," in IEEE Transactions on Human-Machine Systems, vol. 51, no. 5, pp. 524-534, Oct. 2021, doi: 10.1109/THMS.2021.3097973.
[4] Rodríguez, L. M., & Cox, C. (2023, March). Speech-to-text recognition for multilingual spoken data in language documentation. In Proceedings of the Sixth Workshop on the Use of Computational Methods in the Study of Endangered Languages (pp. 117-123).
[5] Henze, N., Löcken, A., Boll, S., Hesselmann, T., & Pielot, M. (2010, December). Free-hand gestures for music playback: deriving gestures with a user-centred process. In Proceedings of the 9th international conference on Mobile and Ubiquitous Multimedia (pp. 1-10).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)