Developing a hands-free interface to operate a Computer using voice command

Mohammad Liton Hossain[*], **Md. Noushad Rana**[**]

[*] Department of ECE, Institute of Science and Technology
[**] Department of CSE, Institute of Science and Technology

DOI: 10.29322/IJSRP.11.03.2021.p11166
http://dx.doi.org/10.29322/IJSRP.11.03.2021.p11166

Abstract- The main focus of this study is to help a handicap person to operate a computer by voice command. It can be used to operate the entire computer functions on the user's voice commands. It makes use of the Speech Recognition technology that allows the computer system to identify and recognize words spoken by a human using a microphone. This Software will be able to recognize spoken words and enable user to interact with the computer. This interaction includes user giving commands to his computer which will then respond by performing several tasks, actions or operations depending on the commands they gave. For Example: Opening /closing a file in computer, YouTube automation using voice command, Google search using voice command, make a note using voice command, calculation by calculator using voice command etc.

Index Terms- Google search, Human-Computer Verbal Interaction, Speech Recognition Technology, Speech Technology, Voice Commands, VAD, Voice Response System.

I. Introduction

This is study focused on controlling the computer by voice command. It is useful for handicap persons who have no ability to operate a computer; they can operate their computer by voice command.

They can browse any website by voice command. This system has included Wikipedia library, so they can collect any information from Wikipedia. They can access files with voice command as like normal persons access file by mouse. Moreover, the system is designed for the handicap persons to carry out operations in a smooth and effective manner. The application is reduced as much as possible to avoid errors while entering the voice command data. It also provides error message while it can't catch voice command. No formal knowledge is needed for the users to use this system. Thus, by this all it proves it is user friendly. This system as described above, can lead to secure and reliable system.

The system will do the following activities:

- Greeting message
- Get voice command
- Voice recognition
- Converting speech to text
- Checking stored data
- Reporting
- Take action

II. Literature Review

Human-computer interfaces facilitate communication, assist on the exchange of information, process commands and controls and perform several additional interactions. Spoken natural language is more user-friendly mean of interacting with a computer. From the human perspective point, this kind of interaction is easier since it does not urge humans to learn additional interactions. Humans can rely on natural ways of communications instead. Human-computer interaction varies from understanding simple commands to extracting all the information in the speech signal such as words, meanings and emotions of the user. To develop an interface with natural language understanding ability, several factors arise and must be taken into account such as dealing with the ungrammatical nature of many spoken utterances, the detection of problems in speech recognition, and the design of intelligent clarification dialogues.

Speech recognition systems can be divided into a number of classes based on their ability to recognize different words. A few classes of speech recognition are classified as under:

- Isolated Speech
  Isolated words usually involve a pause between two utterances; it doesn't mean that, it only accepts a single word, but requires one utterance at a time.
- Connected Speech
  Connected words or connected speech is similar to isolated speech, but allows separate utterances with minimal pauses between them.
- Continuous Speech
  Continuous speech allows the user to speak almost naturally, and is also called computer dictation.

A. How Speech Recognition Works

Early systems were limited to a single speaker and had limited vocabularies of about a dozen words. Modern speech recognition systems have come a long way since their ancient counterparts. They can recognize speech from multiple speakers and have enormous vocabularies in numerous languages.

The first component of speech recognition is, of course, speech. Speech must be converted from physical sound to an electrical signal with a microphone, and then to digital data with an analog-to-digital converter. Once digitized, several models can be used to transcribe the audio to text.

Most modern speech recognition systems rely on what is known as a Hidden Markov Model (HMM). This approach works on the assumption that a speech signal, when viewed on a short enough timescale (say, ten milliseconds), can be reasonably approximated as a stationary process—that is, a process in which statistical properties do not change over time.

In a typical HMM, the speech signal is divided into 10-millisecond fragments. The power spectrum of each fragment, which is essentially a plot of the signal's power as a function of frequency, is mapped to a vector of real numbers known as spectral coefficients. The dimension of this vector is usually small—sometimes as low as 10, although more accurate systems may have dimension 32 or more. The final output of the HMM is a sequence of these vectors

To decode the speech into text, groups of vectors are matched to one or more phonemes—a fundamental unit of speech. This calculation requires training, since the sound of a phoneme varies from speaker to speaker, and even varies from one utterance to another by the same speaker. A special algorithm is then applied to determine the most likely word (or words) that produce the given sequence of phonemes.

One can imagine that this whole process may be computationally expensive. In many modern speech recognition systems, neural networks are used to simplify the speech signal using techniques for feature transformation and dimensionality reduction before HMM recognition. Voice activity detectors (VADs) are also used to reduce an audio signal to only the portions that are likely to contain speech. This prevents the recognizer from wasting time analyzing unnecessary parts of the signal.

III. SYSTEM ANALYSIS AND FEASIBILTY STUDY

A. System Analysis

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements on the system. System analysis is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is viewed as a whole, the inputs are identified and the system is subjected to close study to identify the problem areas. The solutions are given as a proposal. The proposal is reviewed on user request and suitable changes are made. This loop ends as soon as the user is satisfied with the proposal.

B. EXISTING SYSTEM

The current voice command system or assistant for computer are: costly

- It is less user-friendly.
- Not compatible for handicap person
- Single instruction module
- Voice command is complex

C. PROPOSED SYSTEM

The aim of this study is to develop a voice command system that will help to operate a computer. This system is revolutionary to voice control computer system for handicap persons. This system can be used for all type of operation on computer by voice command.

- The system is to eliminate the complexity of operate a computer
- It is user-friendly
- Its voice recognition technology is almost 90% accurate
- Easy to use
- This system is best for handicap persons

D. Feasibility study

Feasibility study of the system is a very important stage during system design. Feasibility study is a test of a system proposal according to its work ability impact on the organization, ability to meet user needs, and effective use of resources. Feasibility study decides whether the system is properly developed or not.

There are five types of feasibility as mentioned below which have been analyzed in this study:

- Technical Feasibility
- Time Schedule feasibility
- Operational feasibility
- Implementation feasibility
- Economic Feasibility

Considering all the feasibility steps this proposed system meets all the feasibility requirements.

IV. PROJECT DESIGN

Software design is the stage in the software engineering process at which an executable software system is developed. For some simple systems, software design is software engineering, and all other activities are merged with this process. However, for large systems, software design is only one of a set of processes (requirements engineering, verification and validation, etc.) involved in software engineering. Software design is a creative activity in which you identify software components and their relationships, based on a customer's requirements. A design is in the programmer's head or roughly sketched on a whiteboard or sheets of paper. Design is about how to solve a problem, so there is always a design process. In design part this study focused on design the data model and software model of a system.

A. *Use Case Diagram of proposed system:*

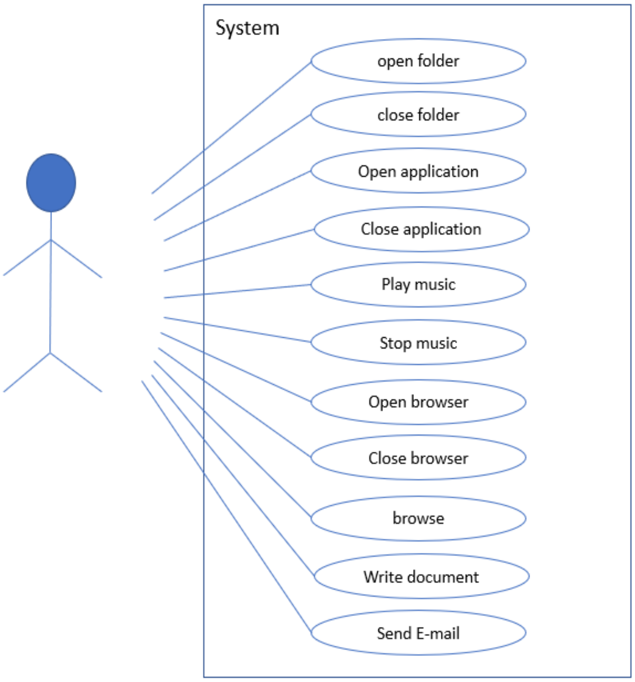


Figure 1: Use case diagram of proposed system

B. *Flowchart of proposed system:*

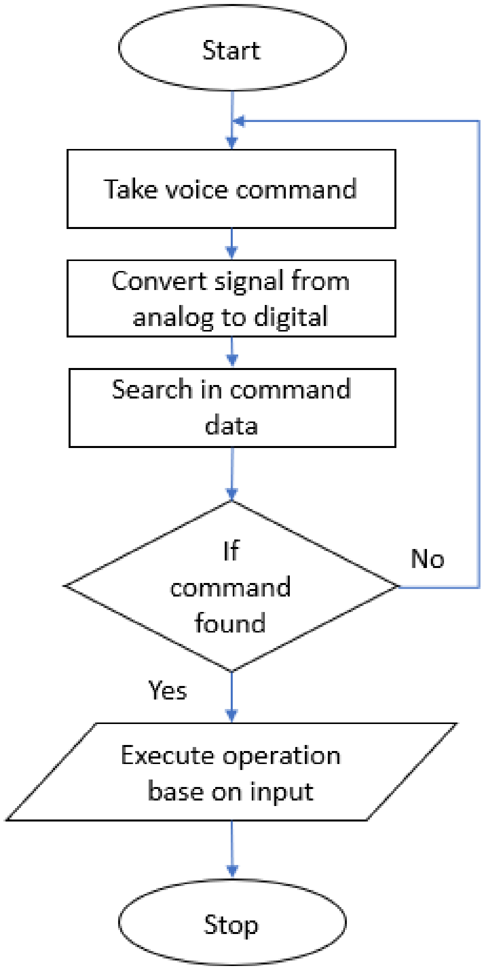


Figure 2: Flowchart of proposed system

C. *Data Flow Diagram (DFD):*

The DFD (Data Flow Diagram) is known as Bubble chart. It is a simple graphical notation that can be used to represent a system in term of the data to the system. Various processing carried out on this data, and the output data generated by system.

- 0-Level DFD:

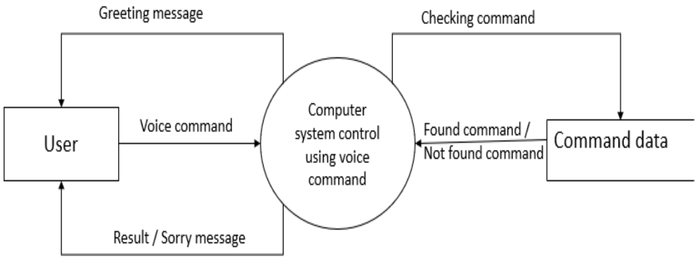


Figure 3: 0-Level DFD

- 1-Level DFD

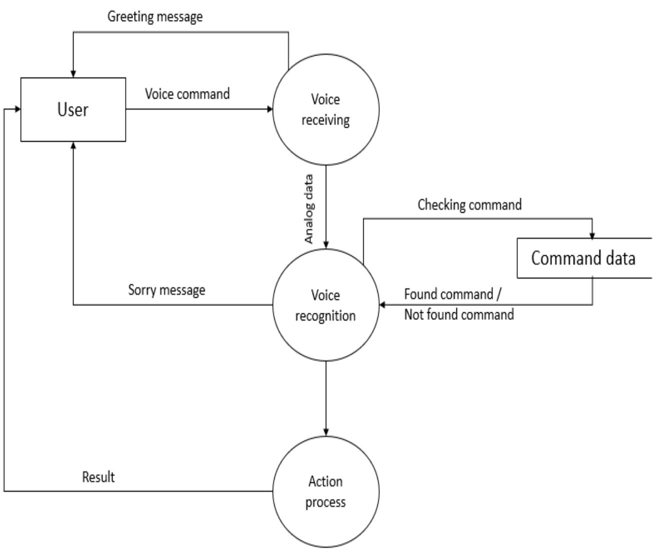


Figure 4: 1-Level DFD

- **2-Level DFD:**

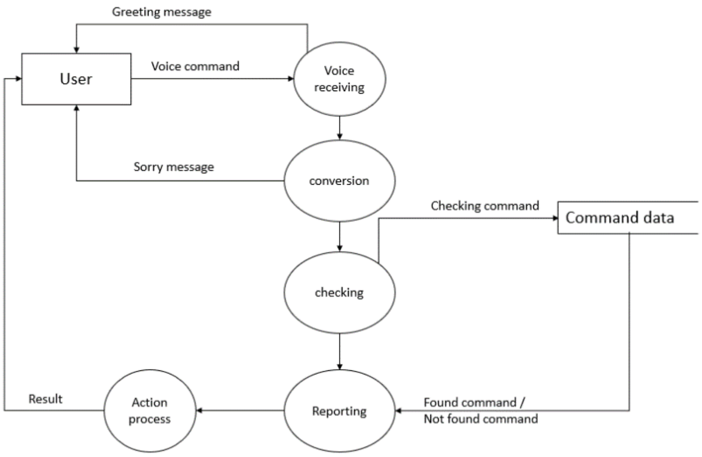


Figure 5: 2-Level DFD

V. Implementation

Software engineering includes all of the activities involved in software development from the initial requirements of the system through to maintenance and management of the deployed system. A critical stage of this process is, of course, system implementation, where this study proposed an executable version of the software. Implementation may involve developing programs in high-level or low-level programming languages or tailoring and adapting generic, off-the-shelf systems to meet the specific requirements of an organization.

A. *Software installation*

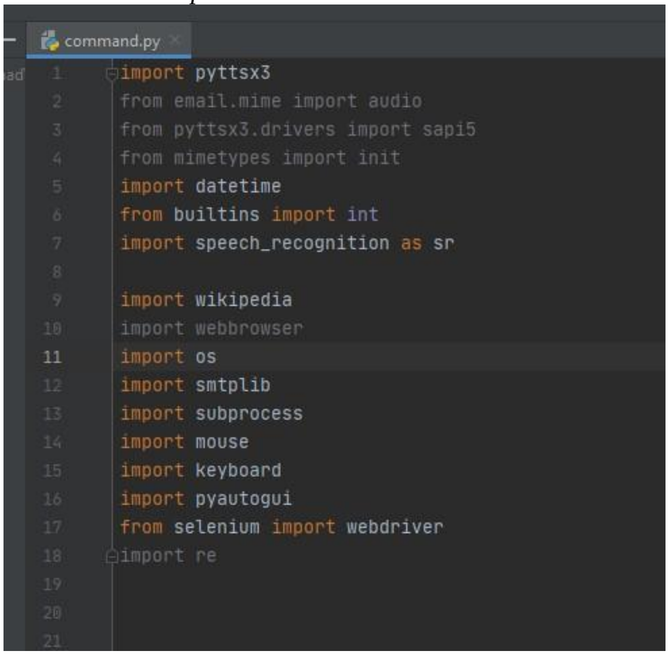
This project used pyCharm IDE, Which is for python programming.

- Python installation
- pyCharm installation

B. *Dependencies installation*

- pyttsx3
- SpeechRecognition
- pyautogui
- Selenium
- wikipedia
- keyboard

C. *Code Implementation*



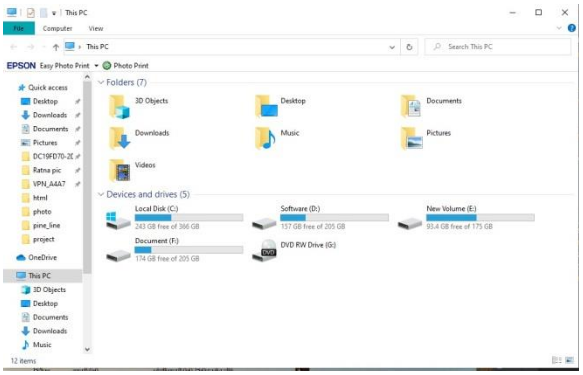
```
import pyttsx3
from email.mime import audio
from pyttsx3.drivers import sapi5
from mimetypes import init
import datetime
from builtins import int
import speech_recognition as sr

import wikipedia
import webbrowser
import os
import smtplib
import subprocess
import mouse
import keyboard
import pyautogui
from selenium import webdriver
import re
```

VI. Testing

Software testing is an activity to check whether the actual result match the expected result and to ensure that the software system is defect free. It involves execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Testing can be both manual and also automation.

A. *Computer Folder Traversing*

- Input Voice Command: “Open My Computer”



http://dx.doi.org/10.29322/IJSRP.11.03.2021.p11166

- Input Voice Command: “Go Down”

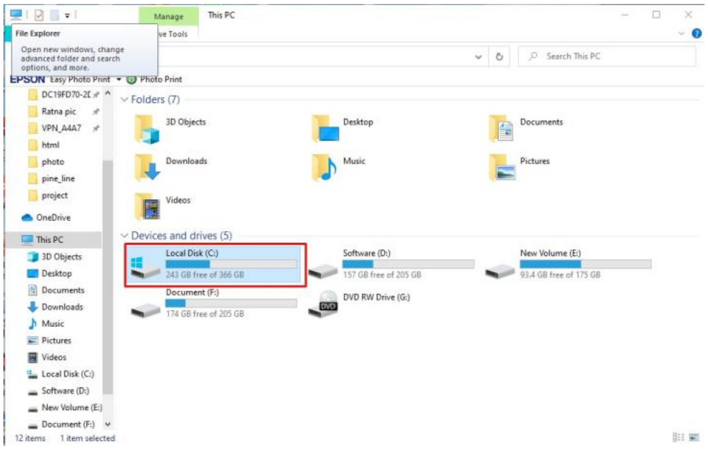
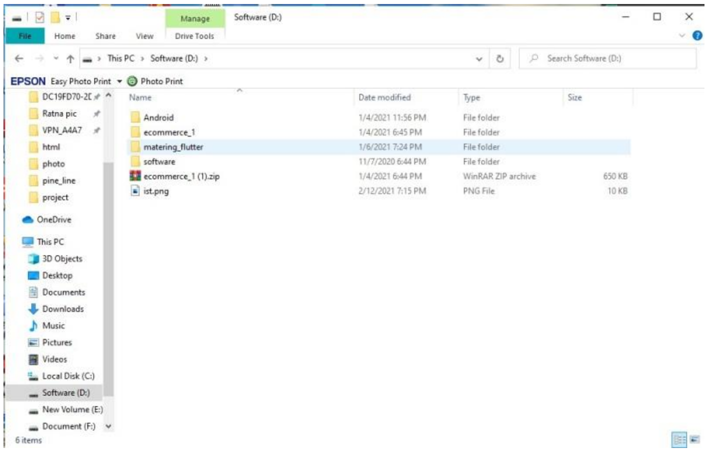


Figure 7: Folder Traversing (Go Down)

- Input Voice Command: “Press Enter”

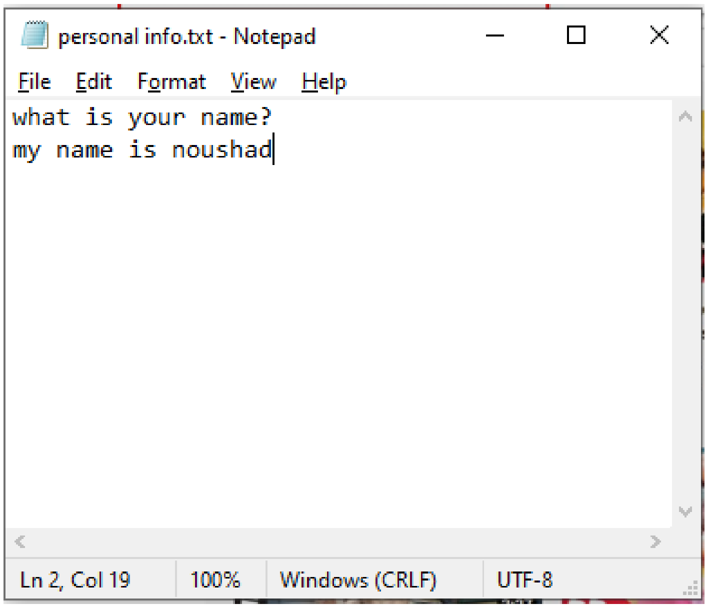


Name	Date modified	Type	Size
Android	1/4/2021 11:56 PM	File folder	
ecommerce_1	1/4/2021 6:45 PM	File folder	
matering_flutter	1/6/2021 7:24 PM	File folder	
software	11/7/2020 6:44 PM	File folder	
ecommerce_1 (1).zip	1/4/2021 6:44 PM	WinRAR ZIP archive	650 KB
ist.png	2/12/2021 7:15 PM	PNG File	10 KB

Figure 8: Folder Traversing (Press Enter)

B. *Making a Note:*

- Input voice command “make a note”
- Output voice “do you want to make a new note! Or open a existing note”
- Input voice command “make a new note”
- Output voice “give a name”
- Input voice command “personal info”
- Input voice command “what is your name”
- Input voice command “my name is noushad”



```
what is your name?
my name is noushad
```

Figure 9: Making a Note

VII. Future Work

This project has enormous potential.

- Integrate AI in this project.
- Make a user friendly UI

VIII. Conclusion

IX. This Proposed system is capable to control computer using voice command. This system used python library which is used for text to speech, speech recognition, web automation, mouse and keyboard control. This system can open my computer root file, traverse any folder in computer, can open browser and can also search in browser. It can make a note as user wish. It can open any exe file and close that file.

References

[1] “voice recognition, definition of”. WebFinance Inc. Archived from the original on 3rd December 2011. Retrieved 21 February 2012.

[2] “voice recognition, definition of”. WebFinance Inc. Archived from the original on 3rd December 2011. Retrieved 21 February 2012.

[3] https://towardsdatascience.cojm m/build-your-first-voice-assistant-85a5a49f6cc1

[4] https://realpython.com/python-speech-recognition/

[5] https://www.geeksforgeeks.org/speech-recognition-in-python-using-google-speech-api/

[6] https://www.python.org/doc/

Authors

First Author – Mohammad Liton Hossain, Assistant Professor, Department of ECE, Institute of Science and Technology (IST)
Email: litu702@gmail.com
Second Author – Md. Noushad Rana, Student, Institute of Science and Technology.

http://dx.doi.org/10.29322/IJSRP.11.03.2021.p11166