



Compsoft technologies-M21

Project Code-CST219166ML23449

Virtual Assistant for Visually Impaired

OVERVIEW

The field of artificial intelligence has led to various virtual assistants such as Siri in iPhone, Google Assistant, Microsoft Cortana, and so on. Even after such progression, very little has been done to implement these technologies to assist the visually impaired community. Recognizing a person or distinguishing an object, these tasks are straightforward for common people but can be very difficult for people that are partly or completely blind. Their lives can be made smoother by assisting them to detect what is present in front of them at that instant. We aim to develop a system/assistant that will serve to guide a visually impaired person and will indicate the person by speaking through the earpiece. The system will help the person recognize people, add new faces and detect objects that are in their vicinity. We will have a mobile application that will consist of numerous deep learning models that will help applications increase their administration. The primary working of the system will consist of the camera continuously feeding images for inputs, the core system processing this input information and the earpiece acting as the output device to provide this output to the user.

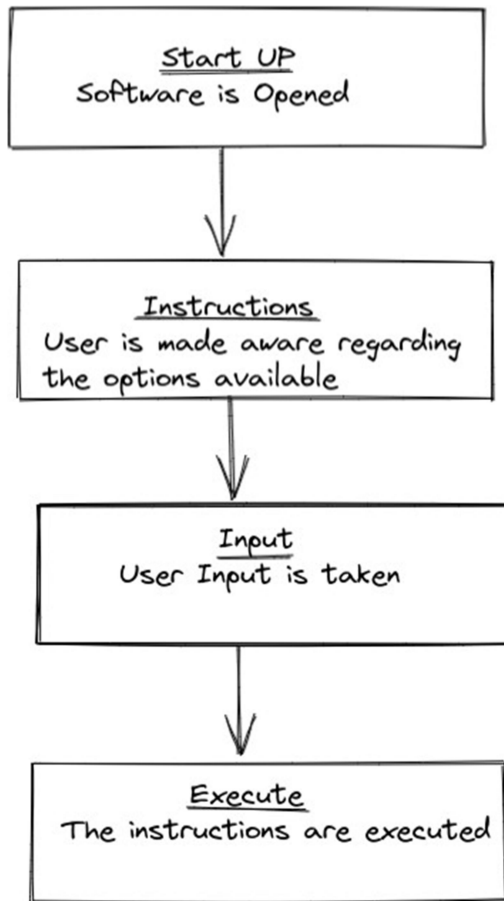
Methodology & Survey

The system comprises a modular client-server distributed architecture. The system consists of the main menu which first runs on the startup of the software and the website modules. The client communicates with the server and back with the use of REST APIs, thus the website modules are not local to the client. Throughout the system, the user communicates with the software via a speech-to-text interface. The Google library of speech-to-text (Speech Recognition) for Python is used for this purpose. For communicating the system's output to the user as well as for confirming the user input, the recognized input is played back to the user using the Python text-to-speech library (pyttsx3). The modules are written in Python and make use of Selenium for automation of the respective module and BeautifulSoup for scraping the contents of the web page. The "Script" component of each module consists of customized code that entails the features of the website contained in the module. For instance, the Wikipedia module consists of a Question and Answer and A summary feature along with the traditional feature of reading out the entire article. The former is implemented by training a BERT model on the Stanford Question Answering Dataset (SQuAD). The APIs that hold the system together are written in Flask. The software is operating system independent to support hassle-free application and usage of the system.

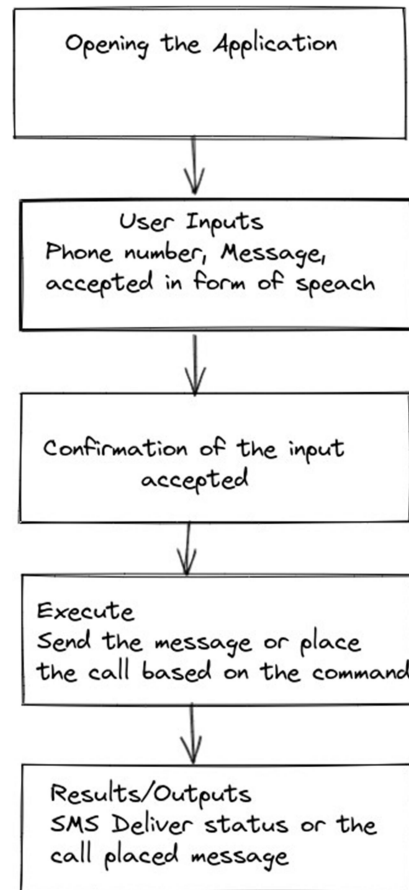
Many researchers have contributed to this field. Various combinations of existing technologies have been used. Braille systems, screen magnifiers, etc. went through some developments but later faced technical issues.

Pilling et al. conducted a study to determine whether the internet provides opportunities for disabled people to carry out activities which they were previously unable to do or whether it leads to greater social exclusion. Sinks and kings et al. state that there is no known research to determine the reasons people with disabilities can't access the internet. Muller et al on the other hand state that the primary barrier to inaccessibility is that of economic and technical capabilities. This thought is seconded by Kirsty et al. who state that bad HTML code and use of pdf causes a hindrance in accessing the internet for the visually impaired. Although the W3C mentions a list of guidelines for maintaining a high level of accessibility for the visually impaired, Power et al. [5] state that only 50.4% of the problems encountered by users were covered by Success Criteria in the Web Content Accessibility Guidelines 2.0 (WCAG 2.0) and 16.7% of websites implemented techniques recommended in WCAG 2.0 but the techniques did not solve the problems.

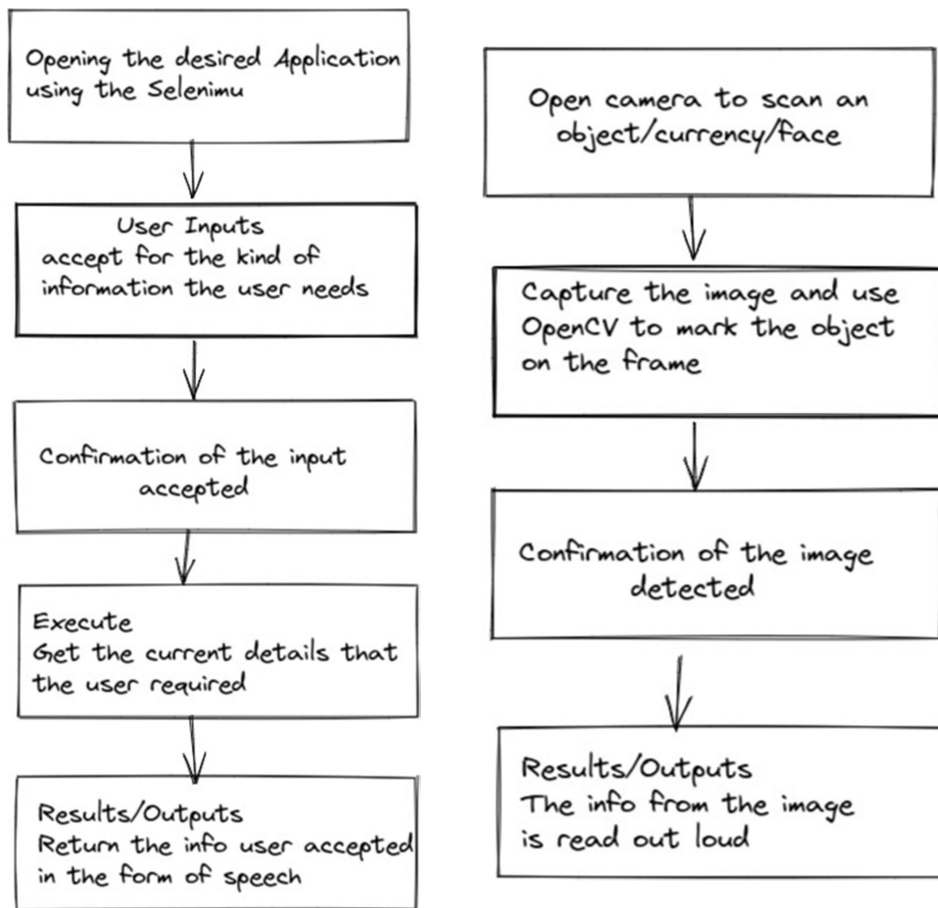
Android phone-controlled voice gesture and touchscreen operated wheelchair where voice and gesture are recognized through android. Developers also created a universal voice control on android which is used to launch android applications via voice commands



Main Menu



Call and Message Module



Time/Date/Location/BatteryLevel/Weather/News/About/Dictionary/CurrencyDetection

Modules to be present(For testing)(any 5)

- Time
- Date
- Location
- Battery Level
- Weather
- Dictionary
- Objects Detection
- Currency Recognition
- Face Recognition
- News
- About
- Message/call

Applications:

Virtual Assistant provides the feature of providing answers to a particular question from a given text of data, thus now the user does not have to read the entire text to figure out the answer, he/she has to simply input the question, the software will find out the answer from the text data on itself using machine learning. The software also provides a summary of the text using machine learning, so the user doesn't have to read the entire thing and thus making it easy to access the website. Thus, using machine learning and speech to text techniques we make the task of accessing the website, which was earlier difficult Now super easy, quick and efficient. Thus, we believe that virtual assistants for the visually impaired are the beginning of Web 3.0

Conclusion:

Various techniques to implement the aforementioned system are analyzed and summarized. Different systems have different ways of implementation along with some limitations and restrictions. These types of systems are very critical for multiple reasons and the occurrence of an error in such a system/device may cause catastrophic damage and loss. The system we are achieving overcomes the limitations of the already implemented systems. Our system consists of a basic UI on a web-based application and comprises several Deep learning models; some of them are object detection, face recognition using TF, TTS, speech recognition and so on. These modules will work together and assist in vital activities like

Src Code:

https://drive.google.com/drive/folders/1cMLkB5coE4_ztgoo9FwGpU45nNFYjLqY?usp=sharing

TASKS to be Completed within 10 days and points list

Important Note : Please note that you need to Earn a Minimum of 35 Points/Complete the mandatory task in order to be eligible to receive a Certificate, Earn more than 60 points, and your project shall be eligible for stipends and other processes.

1. Develop a front end alternate application/Contribute to the existing model on github and Document the same - 30 Points
2. Use the code and existing model that will be provided in this PDF and test its working and compare its accuracy and prepare a report regarding the same- 45 Points [Mandatory task]
3. Improve the accuracy and document the same in detail- 25 Points

Basic tools you will need:

For performing the examples discussed in this tutorial, you will need a Pentium 200-MHz computer with a minimum of 64 MB of RAM (128 MB of RAM recommended).

You will also need the following software –

- Linux 7.1 or Windows xp/7/8/10 or any operating system
- Any IDE(VS Code recommended)

Recommendations:

1. A thorough Theoretical analysis of the topic is required & recommended
2. Use of VSCode is recommended (although the use of any other IDE is okay)
3. Documentation is the key in research

Submission details:

1. You will Create a Project report that Includes Project details, Source code, and Screenshots of Output (word format recommended), the sample report will be shared by the HR team
2. Submission details will be sent later