**ABSTRACT**

Language is man's most important means of communication and speech its primary medium. Spoken interaction both between human interlocutors and between humans and machines is inescapably embedded in the laws and conditions of Communication, which comprise the encoding and decoding of meaning as well as the mere transmission of messages over an acoustical channel. Here we deal with this interaction between the man and machine through synthesis and recognition applications.

Speech recognition, involves capturing and digitizing the sound waves, converting them to basic language units or phonemes, constructing words from phonemes, and contextually analyzing the words to ensure correct spelling for words that sound alike. Speech Recognition is the ability of a computer to recognize general, naturally flowing utterances from a wide variety of users. It recognizes the caller's answers to move along the flow of the call.

Emphasis is given on the modeling of speech units and grammar on the basis of Hidden Markov Model& Neural Networks. Speech Recognition allows you to provide input to an application with your voice. The applications and limitations on this subject enlighten the impact of speech processing in our modern technical field. While there is still much room for improvement, current speech recognition systems have remarkable performance. We are only humans, but as we develop this technology and build remarkable changes we attain certain achievements. Rather than asking what is still deficient, we ask instead what should be done to make it efficient.

**TABLE OF CONTENT**

**Chapter 1: Introduction**

* 1. Introduction
  2. Speech Recognition

**Chapter 2: Literature Survey**

2.1 Speech Recognition Process

2.2 Module Description and Implementation

**Chapter 3: Discussion**

3.1 Speech Recognition Identify

3.2 Validation

**Chapter 4: Testing and debugging**

4.1 Simulated Wave Forms

4.2 Audio Testing

4.3 Distance Testing

**Chapter 5; Google Now Interface**

5.1 SEARCH ENGINE OPTIMIGATION FIRM

5.2 Web Data Mining

5.3 Advantage

5.4 Disadvantage

**Chapter 6: Project Plans**

6.1 Team Structure

6.2 Development Schedule

**7.System Development Life Cycle Model (SDLC Model)**

7.1 System/Information Engineering and Modelling

7.2 System Requirements Analysis

7.3 System Analysis And Design

7.4 Code Generation

7.5 Testing

7.6 Maintenance

.**8. Python**

8.1 What is Python?

8.2 Features of Python

**9. Conclusion & Future Scope**

9.1 Conclusion

9.2 Future Scope

**10. SNAPS**

**ACKNOWLEDGEMENT**

**INTRODUCTION**

**1.1INTRODUCTION**

Have you ever talked to your computer? (And no, yelling at it when your Internet connection goes down or making polite chit-chat with it as you wait for all 25MB of that very important file to download doesn't count). I mean, have you really, really talked to your computer? Where it actually recognized what you said and then did something as a result? If you have, then you've used a technology known as speech recognition.

Speech recognition allows you to provide input to a system with your voice. Just like clicking with your mouse, typing on your keyboard, or pressing a key on the phone keypad provides input to an application, speech recognition allows you to provide input by talking. In the desktop world, you need a microphone to be able to do this.

* 1. **SPEECH RECOGNITION**

The days when you had to keep staring at the computer screen and frantically hit the key or click the mouse for the computer to respond to your commands may soon be a things of past. Today we can stretch out and relax and tell your computer to do your bidding. This has been made possible by the ASR (Automatic Speech Recognition) technology.

Speech recognition is an alternative to traditional methods of interacting with a computer, such as textual input through a keyboard. An effective system can replace, or reduce the reliability on, standard keyboard and mouse input. This can especially assist the following:

* People who have little keyboard skills or experience, who are slow typists, or do not have the time or resources to develop keyboard skills.

1. **LITRETURE SURVEY**
   1. **SPEECH RECOGNITION PROCESS**

Audio

Processing

Distance

Memory

Control

Unit

Register

Display

Voice

Feature\_compar

Extracted

Features

Extract

Write

Address

Data\_out

Enable

Enable

Done

Distance

Synchronizer

Add\_User

ID

Reset

To

(

all

blocks)

Action

Reset\_Reg

2

User

Line\_Count

Pixel\_Count

RGB

Voice

Add

\_User

Reset

* 1. **MODULE DESCRIPTION AND IMPLEMENTATION**
* **OS-win Module**

os-win is released on Pypi, meaning that it can be installed and upgraded via pip. To install os-win, run the following command:

pip install os-win

To upgrade os-win, run the following command:

pip install -U os-win

Note that the first OpenStack release to use os-win is Mitaka. Previous releases do not benefit from this library.

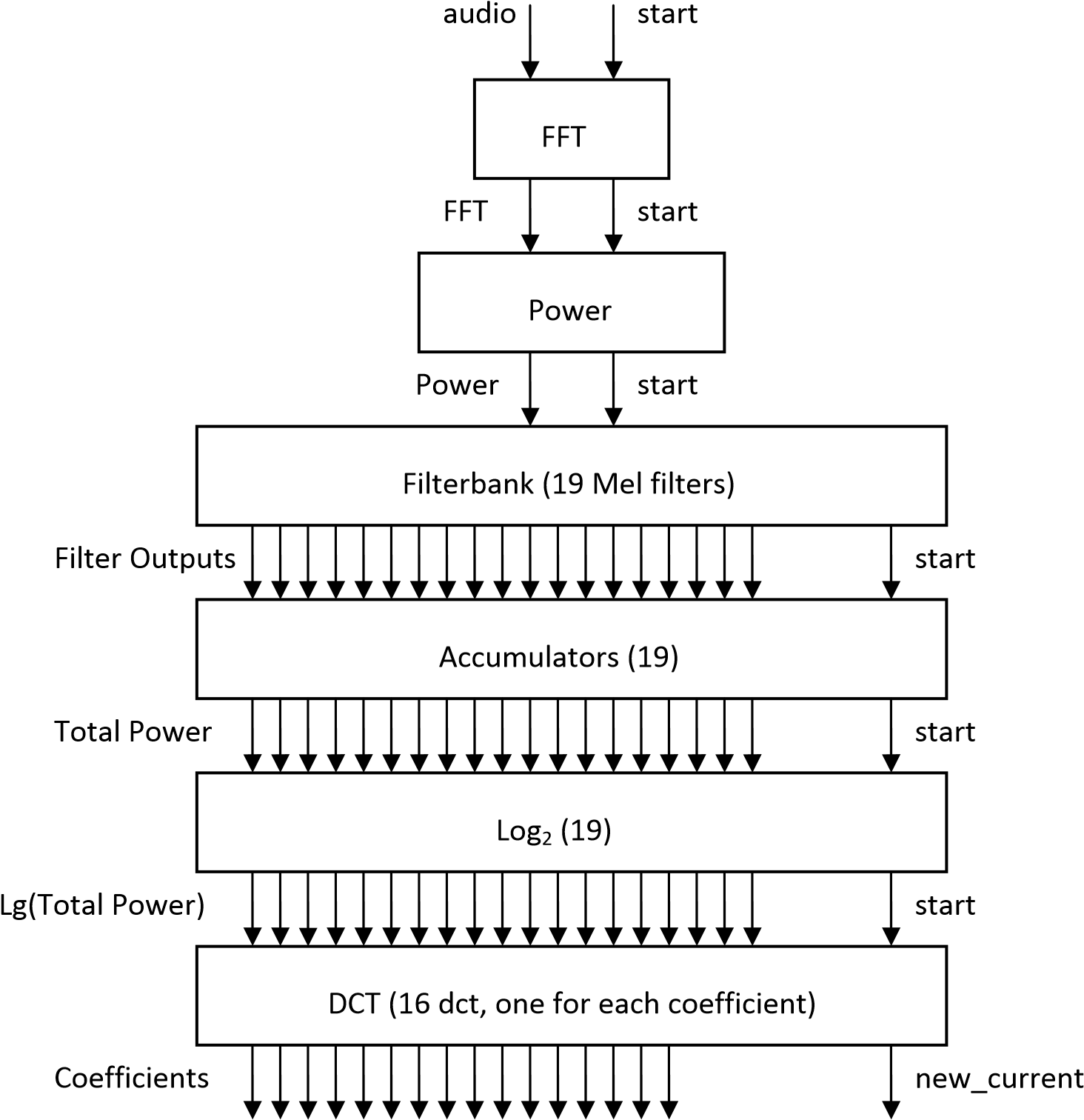
* **PyAudio**

PyAudio provides Python bindings for PortAudio, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms. PyAudio is inspired by:

* pyPort Audio/fastaudio: Python bindings for PortAudio v18 API.
* tkSnack: cross-platform sound toolkit for Tcl/Tk and Python.
* **Wikipedia**
* Python provide a module Wikipedia API that is used to extract wikipedia data.
* The main goal of **Wikipedia-API**is to provide simple and easy to use API for retrieving informations from Wikipedia.
* It supports many operations like extracting text, links, contents, summaries etc from wikipedia.

**BACK-END PROCESS**

After having established correct timing for our input audio, this unit processes the audio data by passing it through the serial network of modules, as shown in Figure.



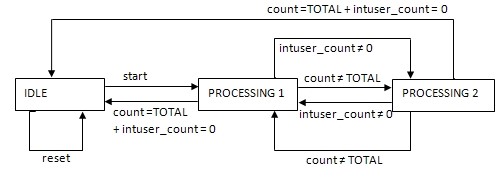
Block Diagram of the Back End Processing Unit

An important design aspect that can be inferred from the architecture is that modules are not arranged in a typical Major-Minor FSM structure. The implemented architecture is based on an all-Minor FSM structure. Such arrangement is not optimal in typical digital systems because it provides less modularity. However, a typical Major-Minor FSM setup for this particular application would not improve modularity either because each data processing sequence is the same. Each block’s function in Figure 4 is self-explanatory and detailed explanation of each is beyond the scope of this report, although it must be mentioned that all blocks were implemented manually, with the exception of the FFT which was obtained from a built-in CoreGen.

### **DISCUSSION**

### **3.1IDENTIFY**

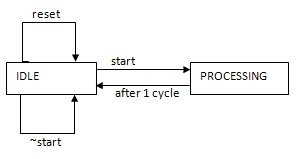
The MEM\_ID module is another minor FSM instantiated by the Control Unit. This module has three stages: IDLE, PROCESSING1, and PROCESSING2. Figure 9 provides a representation of these states. When the Control Unit send a pulse via the id\_start signal, the MEM\_ID module transitions from the IDLE state to the PROCESSING1 state. While transitioning, the module will produce the memory address signal, and will start [fetching] the stored features from the memory. When the module receives the features from the memory, it sends the data to the Distance module.



MEM\_ID Minor FSM State Diagram

* 1. **VALIDATION**

The VALIDATION module is another minor FSM instantiated by the Control Unit. This module has two states: IDLE, and PROCESSING. Figure 10 provides a representation of these states. When the Control Unit sends a pulse via the validation\_start signal, the VALIDATION module transitions from the IDLE state to the PROCESSING state. At the PROCESSING state, the module compares the user’s distances provided by the distance module and selects the user with the smallest distance



VALIDATION Minor FSM State Diagram.

# **4. TESTING AND DEBUGGING:**

Overall, Model Sim simulations were the preferred method of testing individual modules because it is much quicker. However, since total system integration was not possible, to actually prove that the implementations were correct, the subsystems were also tested with FPGA implementations using additional modules that simulated.

### **4.1 SIMULATED WAVEFORMS (R)**

The Control Unit and the minor FSMs were tested by creating Verilog test benches within the project itself. ModelSim generated waveforms that were utilized by the programmer to study the behavior and functionality of the modules. Test benches were created for testing the Control Unit, the Add FSM, the Identification FSM, the Validation FSM, the Memory, and the Action Register.

### **4.2 AUDIO TESTING (J)**

In the case of the Audio Processing subsystem, this was not necessary because the input audio already came from the FPGA, although the start signal was simulated as a standard input button that we could press. The proper output was then analyzed using the digital oscilloscope and appropriate triggering methods. From these tests it was discovered that the actual outputs on the accumulators were not what they should be although the path up to the filter outputs seemed fine. The problem could not be solved.

### **DISTANCE TESTING (J)**

Meanwhile, to test the Distance Processor subsystem, a similar approach was taken, a test bench that simulated constant input audio spectral coefficients and two different sets of constants stored coefficients was generated. The module that generated this is included in the Appendix. Upon pressing the start input button, a state toggled and one of two stored sets of spectral coefficients was chosen. This remained constant throughout the whole sequence. The test bench then cycled through all the necessary start/ready loops and finally signaled the processor that no more data was stored. The observed output showed correct values for each of the two sets of stored coefficients.

**5 GOOGLE NOW INTERFECE**

**5.1 SEARCH ENGINE OPTIMIGATION FIRM**

It offers the facility of Optimizing and Positioning of your website in the Major Search Engines. They initiate the **SEO Service** process by determining the Keyword/Phrase that best describe your Website/Business. Then they build META Tags, for the few search engines that still use these.

The placement and maintenance of your website is monitored through out the year. This becomes even more important whenever the search engines changes their specifications. Maintenance is an extremely important aspect of a quality web site, not just for the benefit of search engines, but also for overall accuracy of your company's information.

**5.2.WEB DATA MINING**

It is a proven technology for advanced analysis that detects key patterns and trends. But the time-consuming complexity of preparing Web data with the business context necessary for data mining has hampered its use in Web analysis—until now.

EME Technologies creates, maintains, and runs Internet robots that retrieve data from the Web. The robots feed extractors that pick out useful information and can deliver it to you in a format for processing and analysis.

**5.3 ADVANTAGES**

It is helpful for those people who don’t want to use keyboard.It is less time consuming than keyboard. Anyone can operate easily.

**5.4 DISADVANTAGES**

Due to large of data a lot of complexities are involved in maintaining, updating and retrieving selected information, since old system is totally maintained manually, some of the complexities involved in existing system is as follows:

* **Redundancy of data:** In the existing system, all data is maintained in different files and registers. This often leads to considerable duplication of data and redundancy.
* **Difficulty in** **Updating the Data:** Problem of updating data in the existing systems since everything is stored in registered and files. It is very difficult and time consuming to update data

**6. PROJECT PLANS**

**6.1 TEAM STRUCTURE**

I did this project alone.

* Our very able and helpful project guide Mr. Subhdeeep.

**6.2DEVLOPMENT SCHEDULE**

The work in developing the new system commented immediately with our first meeting with the management and users thereafter we were in continuous touch with the management and users of the company.

**7. System Development Life Cycle Model (SDLC Model)**

This is also known as Classic Life Cycle Model (or) Linear Sequential Model (or) Waterfall Method. This has the following activities.

1. System/Information Engineering and Modeling

2. Software Requirements Analysis

3. Systems Analysis and Design

4. Code Generation

5. Testing

6. Maintenance

**7.1 SYSTEM/INFORMATION ENGINEERING AND MODELING**

As software is always of a large system (or business), work begins by establishing requirements for all system elements and then allocating some subset of these requirements to software. This system view is essential when software must interface with other elements such as hardware, people and other resources. System is the basic and very critical requirement for the existence of software in any entity. So if the system is not in place, the system should be engineered and put in place. In some cases, to extract the maximum output, the system should be re-engineered and spruced up. Once the ideal system is engineered or tuned, the development team studies the software requirement for the system.

**7.2 SYSTEM REQUIREMENT ANALYSIS**

This is also known as feasibility study. In this phase, the development team visits the customer and studies their system. They investigate the need for possible software automation in the given system. By the end of the feasibility study, the team furnishes a document that holds the different specific recommendations for the candidate system. It also includes the personnel assignments, costs, project schedule, and target dates. The requirements gathering process is intensified and focused specially on software. To understand the nature of the program(s) to be built, the system engineer ("analyst") must understand the information domain for the software, as well as required function, behavior, performance and interfacing. The essential purpose of this phase is to find the need and to define the problem that needs to be solved.

**7.3 SYSTEM ANALYSIS AND DESIGN**

In this phase, the software development process, the software's overall structure and its nuances are defined. In terms of the client/server technology, the number of tiers needed for the package architecture, the database design, the data structure design etc are all defined in this phase. A software development model is created. Analysis and Design are very crucial in the whole development cycle. Any glitch in the design phase could be very expensive to solve in the later stage of the software development. Much care is taken during this phase. The logical system of the product is developed in this phase.

**7.4 CODE GENERATION**

The design must be translated into a machine-readable form. The code generation step performs this task. If the design is performed in a detailed manner, code generation can be accomplished without much complication. Programming tools like Compilers, Interpreters, and Debuggers are used to generate the code. Different high level programming languages like C, C++, Pascal, and Java are used for coding. With respect to the type of application, the right programming language is chosen.

**7.5 TESING**

Once the code is generated, the software program testing begins. Different testing methodologies are available to unravel the bugs that were committed during the previous phases. Different testing tools and methodologies are already available. Some companies build their own testing tools that are tailor made for their own development operations.

**7.6 MAINTAINACE**

Software will definitely undergo change once it is delivered to the customer. There are many reasons for the change. Change could happen because of some unexpected input values into the system. In addition, the changes in the system could directly affect the software operations. The software should be developed to accommodate changes that could happen during the post implementation period.

**8.PYTHON**

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, <https://www.python.org/>, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well.

* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

## **PYTHON FEATURES**

Python's features include −

* + **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
  + **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − We can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.

**9. CONCLUSION & FUTURE SCOPE**

**9.1 CONCLUSION**

The objective of this final project was to design and implement a complex digital system combining voice, video and user interfaces. The analysis presented in the previous sections shows a fully functional control units, video displays, user interfaces, and voice extraction system. In addition, it includes a fully functional voice modulation algorithm, and a partially functional voice modulation system. A comprehensive testing and debugging methodology was utilized, which validated the functionality of the different modules.

Overall this project could have been a great success. However, due to time constraints and the complexity of what the team wanted to do impeded the integration of all modules and the completion of our project.

**9.2 FUTURE SCOPE**

**9.2.1 Achieving efficient speaker independent word recognition**

All the SR system will be speaker independent and will produce the same kind output for a particular command irrespective of the user.SR system will be able to process the voice commands of all the user with very high accuracy & efficiency.

**9.2.2 Ability to distuinguish nuances of speech and meanings of words**

SR systems would be able to distuingsh between nuances phrases & meaningful commands & would be able to process the proper command out of the nuances phrases correctly.

**9.2.3 Stand-alone Speech Recognition System**

Presently there is no SR stand alone systems available, all the SR systems been developed are based on or the other preexisting hardware and software platforms.

**9.2.4 Talk with all devices**

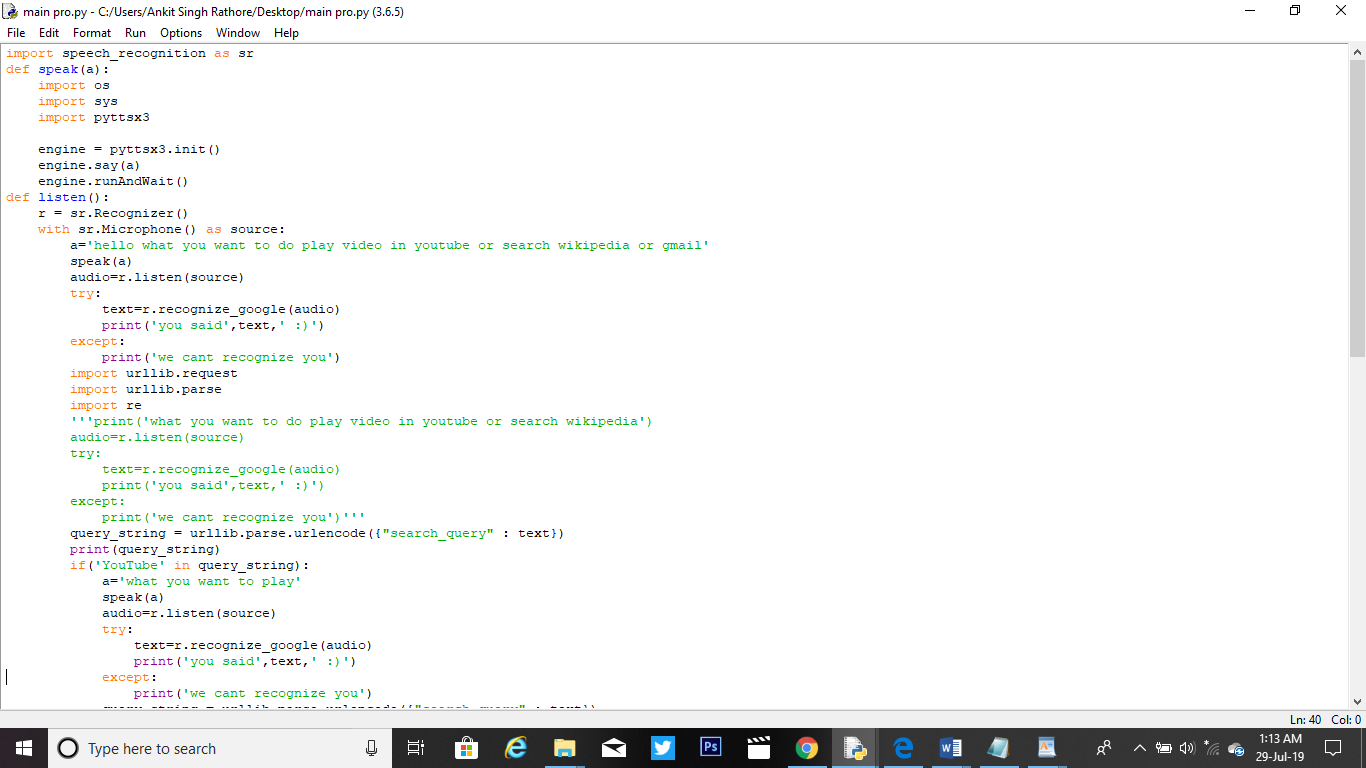
All the devices including Smart phones,Computers, TV, etc will be controlled with voice command of the user .There will be no need of having a Remote or pressing button on the devices to interact with it

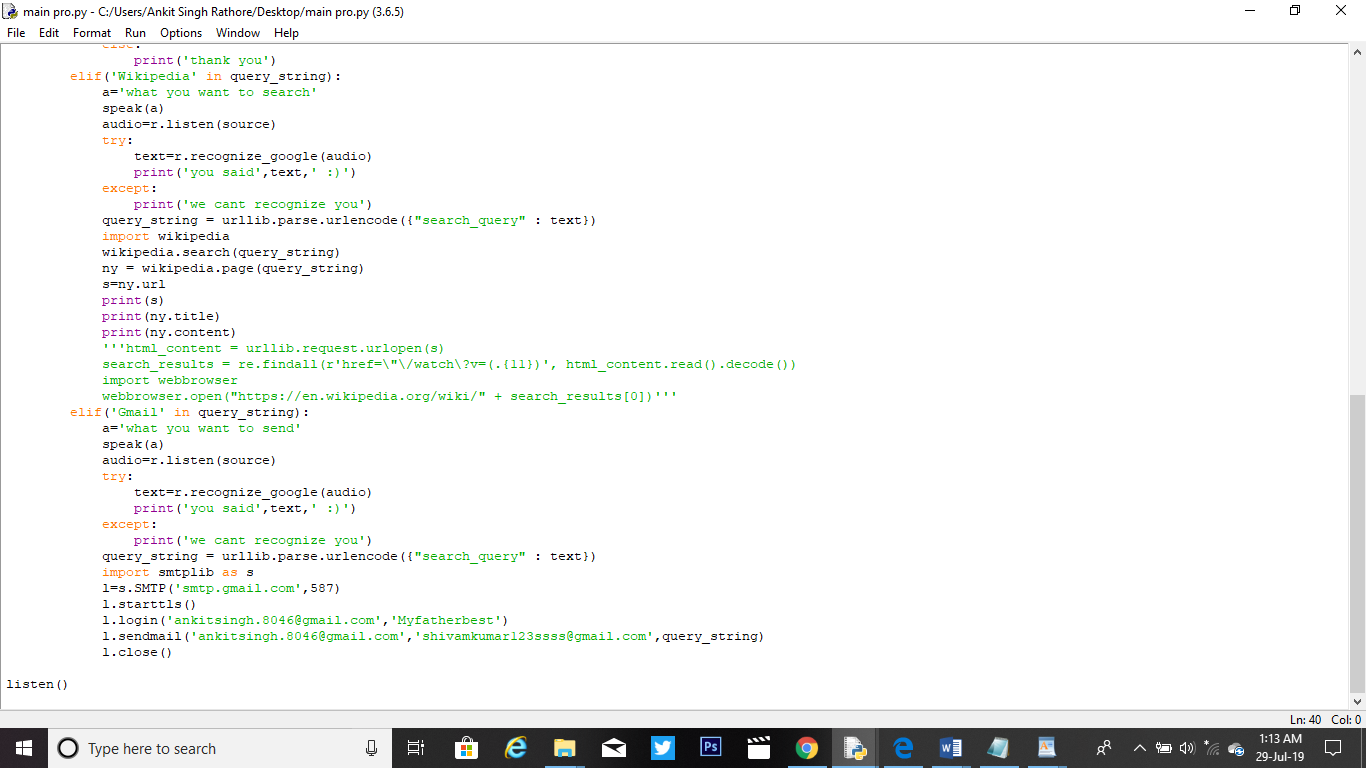
.

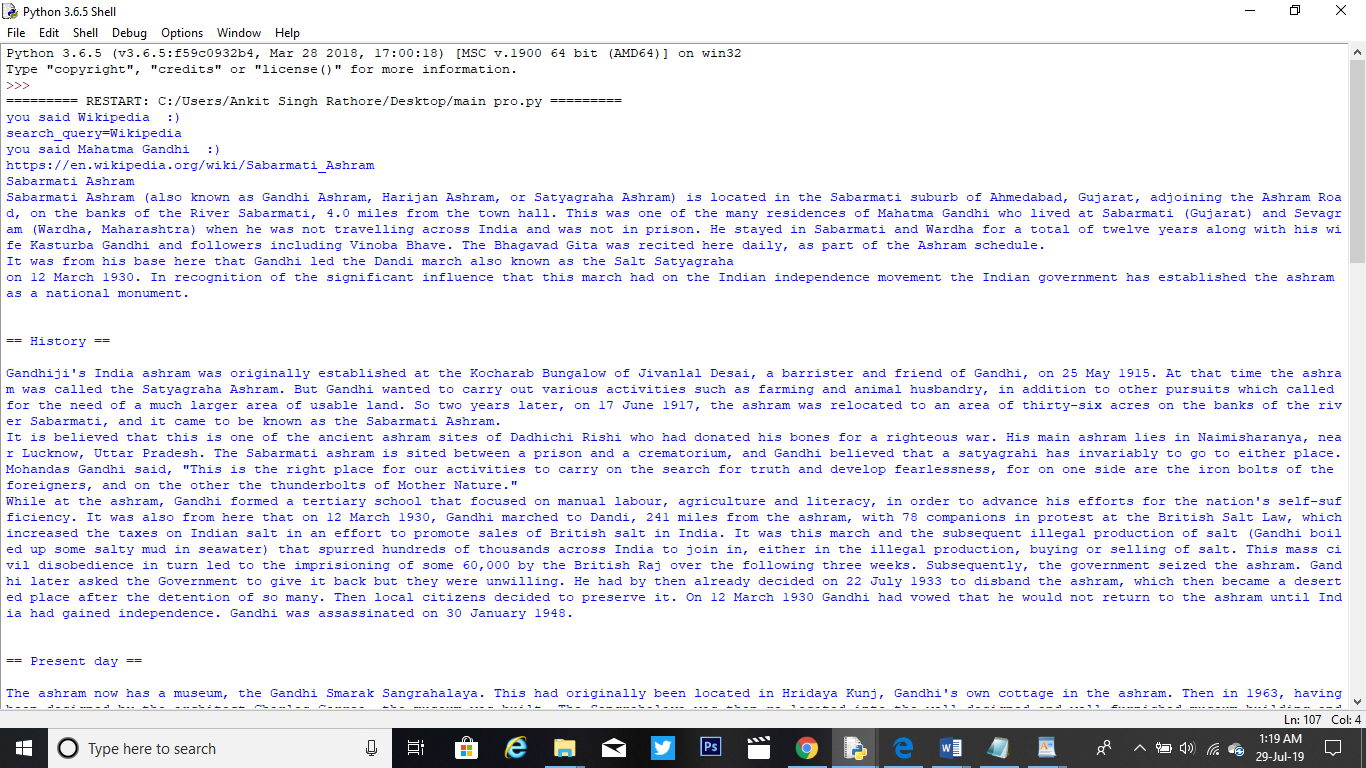
**9.2.5 Wearable Speech Recognition**

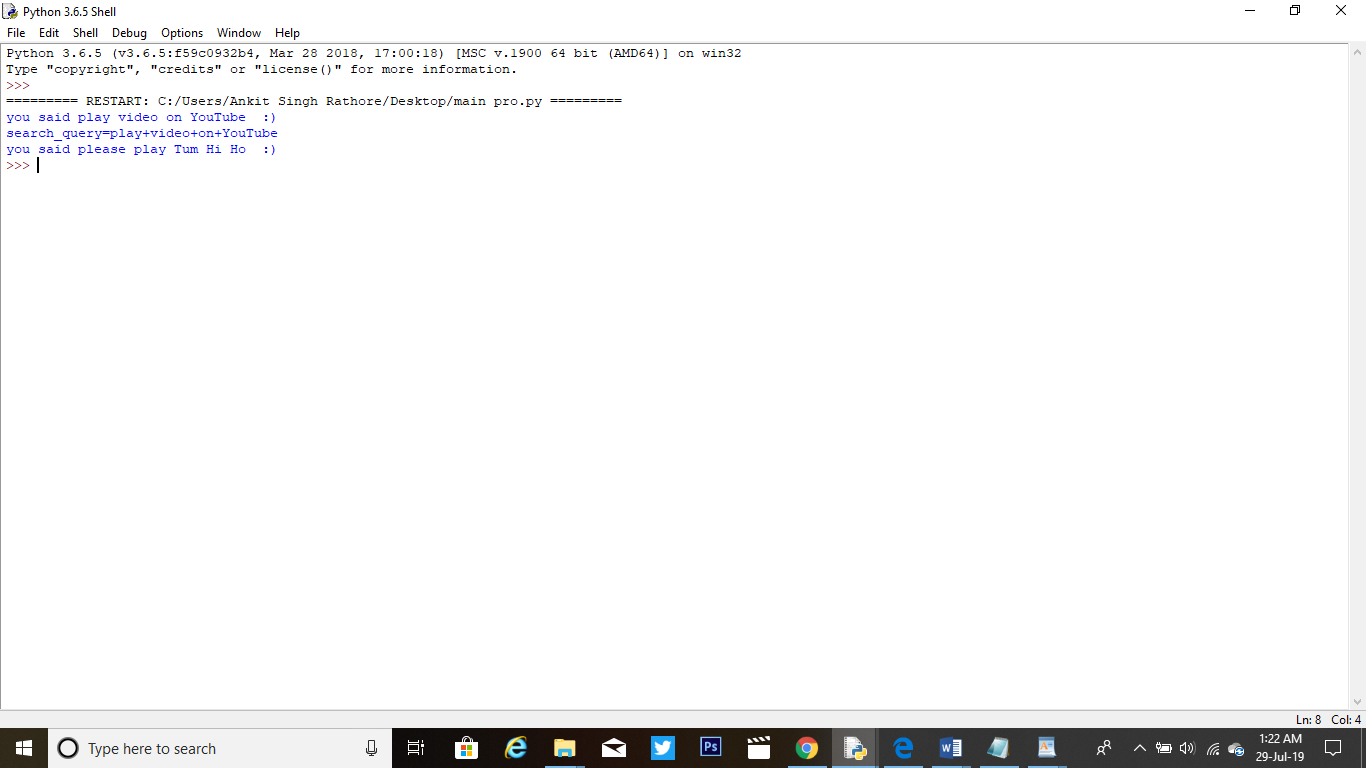
SR systems will be embedded in wearable devices or things such as wrist watch, braclets etc.There will be no need of carriying bulky devices and technologies can be used on the go.

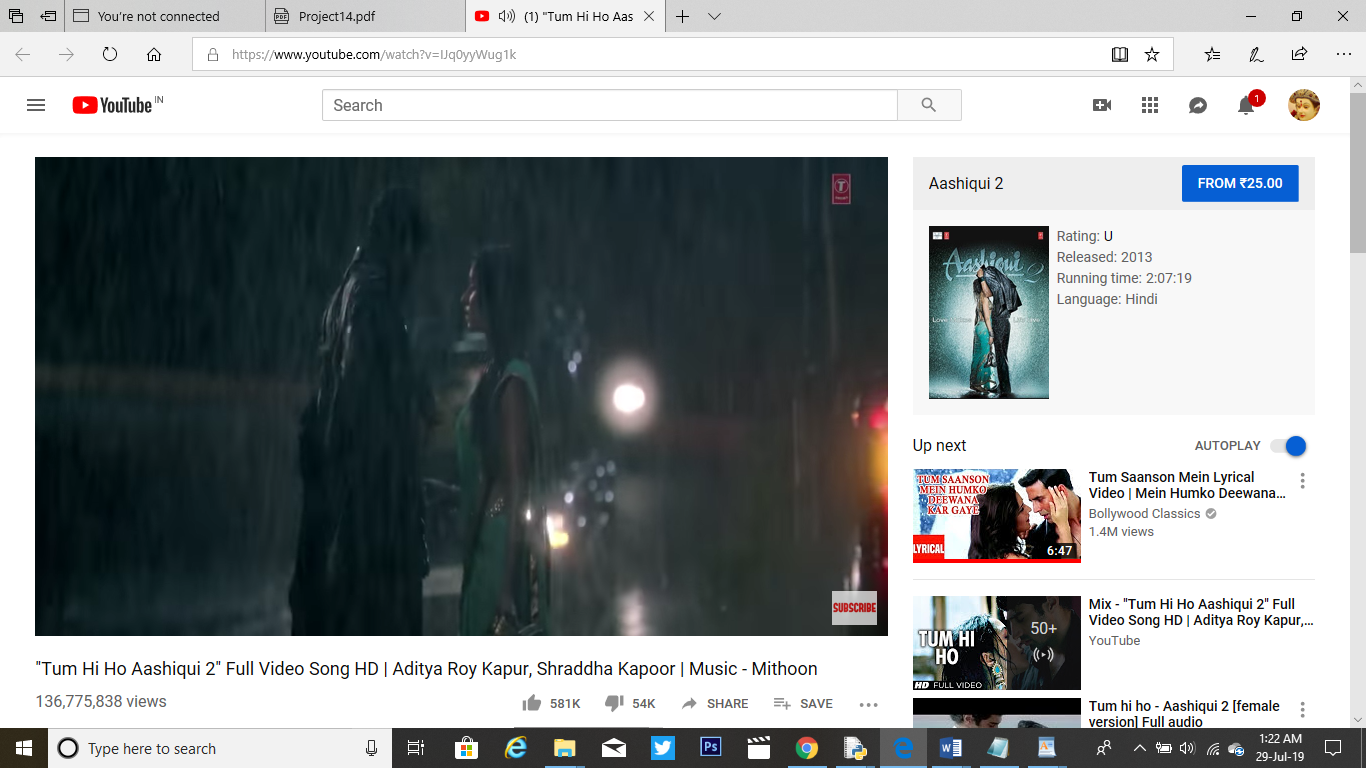
**SNAPS**

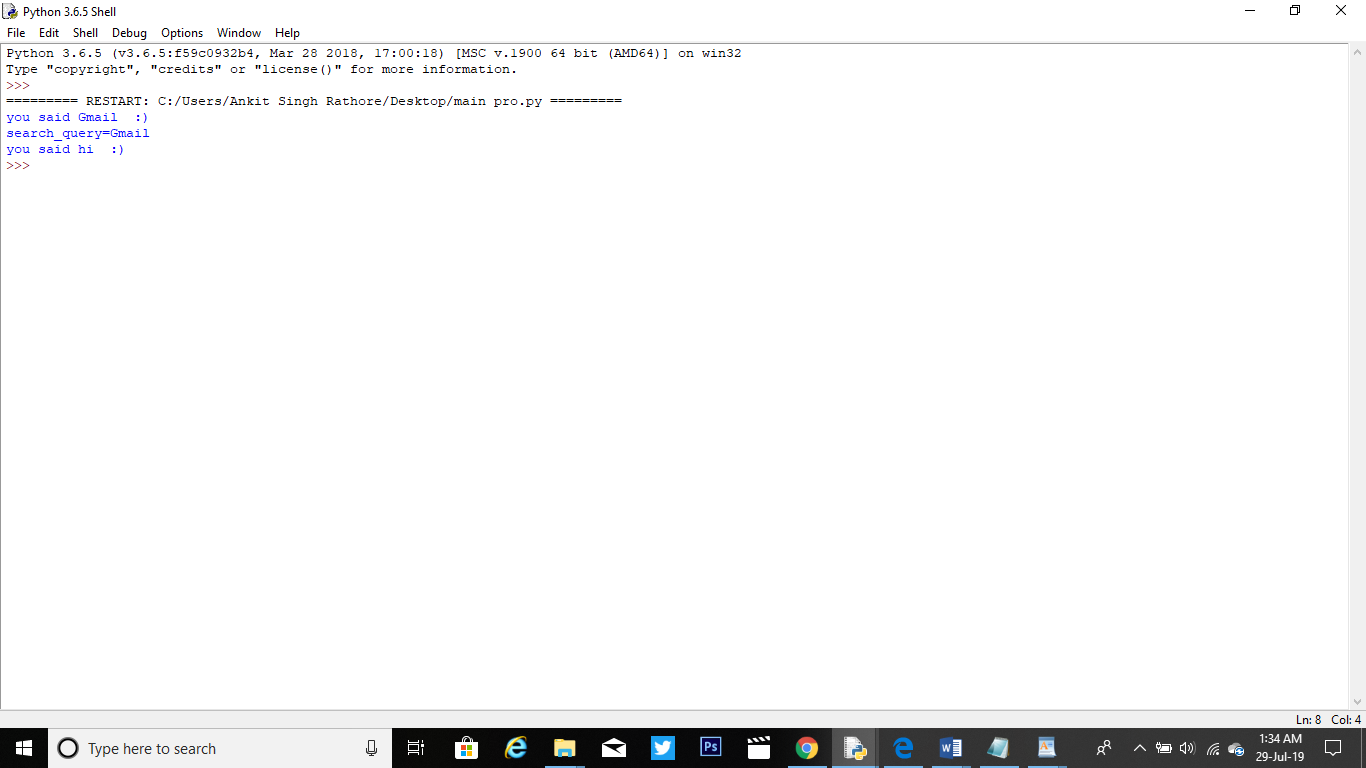


****

****

****

****

****

**ACKNOWLEDGMENT**

**Keep away from people who try to belittle your ambitions. Small people always do that, but the really great make you feel that you too, can become great.**

I take this opportunity to express my sincere thanks and deep gratitude to all those people who extended their wholehearted co-operation and have helped me in completing this project successfully.

First of all, I would like to thank **Mr. Shubhdeep Sir, Director,**  for creating opportunities to undertake me in the esteemed organization.

Special thanks to **Parul Maam** Project Manager for all the help and guidance extended to me by her in every stage during my training. Her inspiring suggestions and timely guidance enabled me to perceive the various aspects of the project in a new light.

I also want to thank my teammates, friends and staff members of BTECH department that have shared their needs and experiences with me.

My report will remain incomplete if I do not make a mention about my parents who expended all moral and financial support to me. I would like to special thanks to my parents.

I would also like to thank my parents and friends, without their continuous motivation, help and support this would not have been possible

**NAME:- ANKIT KUMAR**

**UID:- 17BCS2600**