

A
MINI PROJECT REPORT
ON
SPEECH RECOGNITION USING PYTHON

Submitted in partial fulfilment of the requirements for the award of the
degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted By:

B. BILWANI (21UP1A05E0)

D. PAVANI (21UP1A05E6)

S. NANDINI (21UP1A05H7)

Under the Guidance of

Mrs. B. RAMYA SRI

(Assistant Professor)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VIGNAN'S INSTITUTE OF MANAGEMENT AND TECHNOLOGY
FOR WOMEN

(An Autonomous Institution)

**(Affiliated to Jawaharlal Nehru Technological University Hyderabad,
Accredited by NBA, NAAC with A+)**

Kondapur (Village), Ghatkesar (Mandal), Medchal (Dist.)

Telangana-501301

(2021-2025)



**VIGNAN'S INSTITUTE OF MANAGEMENT AND
TECHNOLOGY FOR WOMEN**
(An Autonomous Institution)

[Sponsored by Lavu Educational Society, Affiliated to JNTUH & Approved by AICTE, New Delhi]
Kondapur (V), Ghatkesar (M), Medchal - Malkajgiri (D) - 501 301. Phone: 96529 10002/3



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work entitled “**SPEECH RECOGNITION SYSTEM USING PYTHON**” submitted by **B. BILWANI (21UP1A05E0), D. PAVANI (21UP1A05E6), S. NANDINI (21UP1A05H7)** in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering, Vignan's Institute of Management and Technology for Women** is a record of Bonafide work carried by them under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or institute for the award of any degree.

Project Guide:

Mrs. B. RAMYA SRI

(Assistant Professor)

The Head of Department:

Mrs. M. PARIMALA

(Associate Professor)

(External Examiner)



**VIGNAN'S INSTITUTE OF MANAGEMENT AND
TECHNOLOGY FOR WOMEN**
(An Autonomous Institution)

[Sponsored by Lavu Educational Society, Affiliated to JNTUH & Approved by AICTE, New Delhi]
Kondapur (V), Ghatkesar (M), Medchal - Malkajgiri (D) - 501 301. Phone: 96529 10002/3



DECLARATION

We hereby declare that the results embodied in the project **entitled “SPEECH RECOGNITION SYSTEM USING PYTHON”** is carried out by us during the year 2024-2025 in partial fulfillment of the award of **Bachelor of Technology** in **Computer Science and Engineering** from **Vignan's Institute of Management and Technology for Women** is an authentic record of our work under the guidance of Guide Name. We have not submitted the same to any other institute or university for the award of any other Degree.

B. BILWANI (21UP1A05E0)

D. PAVANI (21UP1A05E6)

S. NANDINI (21UP1A05H7)

ACKNOWLEDGEMENT

We would like to express sincere gratitude to **Dr. G. AppaRao Naidu, Principal, Vignan's Institute of Management and Technology for Women** for his timely suggestions which helped us to complete the project in time.

We would also like to thank our sir **Mrs. M Parimala, Head of the Department, Computer Science and Engineering**, for providing us with constant encouragement and resources which helped us to complete the project in time.

We would like to thank our project guide, **Mrs. B. Ramya Sri, Assistant Professor of VMTW, Computer Science and Engineering**, for her timely cooperation and valuable suggestions throughout the project. We are indebted to her for the opportunity given to work under her guidance.

Our sincere thanks to all the teaching and non-teaching staff of Department of Computer Science and Engineering for their support throughout our project work.

B. BILWANI (21UP1A05E0)

D. PAVANI (21UP1A05E6)

S. NANDINI (21UP1A05H7)

INDEX

S. No	Topic	Pg.no
	ABSTRACT	1
1.	INTRODUCTION	2-10
	1.1 What is speech recognition	3
	1.2 Motivation	4-5
	1.3 Existing System	6
	1.4 Challenges in Existing System	6-7
	1.5 Proposed System	8
	1.6 Advantages of the Proposed System	9
	1.7 Objectives	9
	1.8 Methodology	10
2.	LITERATURE SURVEY	11-12
3.	SYSTEM ANALYSIS	13-23
	3.1 Purpose	13
	3.2 Scope	14-15
	3.3 Feasibility Study	16-22
	3.3.1 Economic Feasibility	16
	3.3.2 Technical Feasibility	17-18
	3.3.3 Social Feasibility	18-19
	3.4 Requirement Analysis	19-22
	3.4.1 Functional Requirements	19-20
	3.4.2 Non-Functinal Requirements	21-22
	3.5 Requirement Specifications	23
	3.5.1 Hardware Requirements	23
	3.5.2 Software Requirements	23
	3.5.3 Language Specification	23

4.	SYSTEM DEVELOPMENT/DESIGN	24-30
	4.1 Speech Synthesis	24
	4.1.1 Evaluation of synthetic speech	24
	4.1.2 Building speech synthesis systems	24
	4.2 Packages used	25
	4.3 System Architecture	26
	4.4 UML Diagrams	27
	4.4.1 Use Case Diagram	27
	4.4.2 Activity Diagram	28
	4.4.3 Sequence Diagram	29
	4.4.4 Class Diagram	30
5.	IMPLEMENTATION AND RESULTS	31-32
	5.1 Methods/Algorithms Used	31
	5.2 Sample Code	32
6.	SCREENSHOTS	33-37
7.	SYSTEM TESTING	38
8.	CONCLUSION	39
9.	FUTURE SCOPE	40
10.	BIBILIOGRAPHY	41
	10.1 References	

LIST OF FIGURES

S.no	Figure Names	Pg.No
1	Fig.1: speech recognition using python	2
2	Fig.2: speech to text conversion	3
3	Fig 3: speech recognition	5
4	Fig.4: Sound waves	7
5	Fig.5: Speech recognition using python	10
6	Fig 6:system development	25

ABSTRACT

One of the fastest-growing engineering innovations is speech recognition. This has been planned and built with that fact in mind, and some effort has been made to accomplish this goal. It has a variety of uses and possible benefits in a variety of fields. Nearly 20% of the world's population has some kind of disability, with a huge number of them being unseeing or incapable of properly handle their arms and some persons who are blind but have difficulty examination difficulties may listen to a researched article using an accessible device. In such situations, speech recognition systems come in-hand, allowing them to exchange information with others when running a device using voice input. A speech to text system (STT) converts speech into text in a human language format and text to speech system (TTS) translates text into speech in a human language format. The proposed device is a hardware solution for synthesizing speech and allowing voice access to digital content. In the modern era, the current technological development provides greater facilities for human life. Speech recognition system (ASR) is improved a lot from the early 1900's to present it allows computer to understand human language speech recognition is a machines ability to listen to spoken words and identify them. You can then use speech recognition in python to convert the convert the spoken words into text make a query or give reply. You can even program some devices to respond to these spoken words. This system also recognizes the error between the given inputs and corrects them accordingly for easier communication. The time taken for correcting errors it would be between 0-15 or 20seconds.

CHAPTER 1

INTRODUCTION

There is huge development in speech recognition technologies from last years as it had completely brought up huge progress based on the new machine learning algorithms. The speech recognition system proves to be beneficial in many aspects as it reduces the wastage of time as well as helps the disabled individuals.



Figure-1: speech recognition using python

Speech technology with fields within the scope of the paper are to be presented in Fig. as the unified framework that encompasses covered topics, showing their complementarity, ranges and borders, interconnections, and intersections in the interdisciplinary area of Speech.

In mostly areas of the country, there are lot of people who don't know how to write and also how to read any word, so this project is very helpful for these type of people as you know in today's world Everybody has its own mobile phones and they want to search a lot of things. In this project, they usually speak what they want to search, and various results of such type opens in the browser window.

1.1 WHAT IS SPEECH RECOGNITION?

Speech recognition incorporates computer science and linguistics to identify spoken words and convert them to text. It allows a computer to understand human languages. Speech recognition is a machine's ability to listen to spoken words and identify them. You can then use speech recognition in Python to convert the spoken words into text, make a query or give a reply. You can even program some devices to respond to these spoken words.

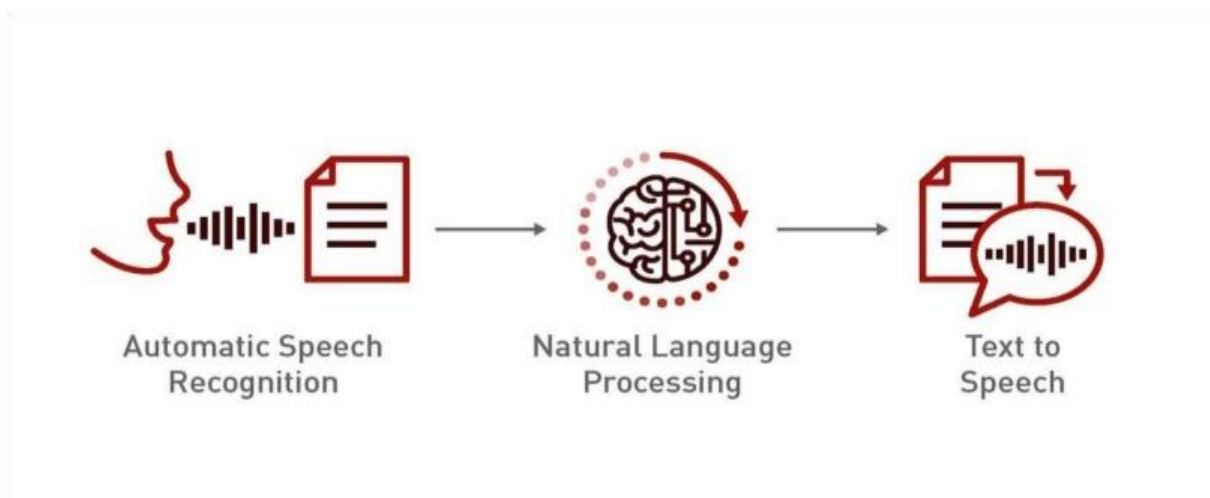


Fig.2 Speech to Text conversion

Step1. Initially audio is taken as input with the help of microphone.

Step2. In next step the audio input is passed through the Natural Language Processing. This helps in text processing, language understanding, and text generation. Making it easier to build robust and accurate speech recognition applications.

Step3. In the final step the voice /speech is converted into written text.

1.2 Motivation for Speech Recognition Using Python:

Accessibility and Inclusivity:

Speech recognition technology can make digital content more accessible to individuals with disabilities, such as those with visual impairments or mobility issues. By enabling voice commands and dictation, users can interact with technology more easily and efficiently.

Improved User Experience:

Integrating speech recognition into applications can significantly enhance the user experience by providing a hands-free and intuitive method of interaction. This can be particularly useful in scenarios where typing is inconvenient or impossible, such as driving or cooking.

Efficiency and Productivity:

Speech recognition can speed up tasks that would otherwise require manual input. For instance, professionals can use speech-to-text for quickly drafting emails, creating documents, or setting reminders, thus improving productivity and saving time.

Technological Advancement and Innovation:

Developing speech recognition systems in Python contributes to the advancement of natural language processing (NLP) and artificial intelligence (AI). It pushes the boundaries of what machines can understand and respond to, fostering innovation in various fields such as customer service, healthcare, and education.

Customizability and Flexibility:

Python offers a rich ecosystem of libraries and frameworks (like Speech Recognition, PyDub, and Deep Speech) that make it relatively easy to develop, customize, and extend speech recognition applications.

Research and Development:

Speech recognition is a vibrant area of research in AI and machine learning. Working on such projects can contribute to academic and commercial research, helping to solve complex problems related to language understanding, accent variation, noise reduction, and more.

Market Demand and Career Opportunities:

There is a growing demand for voice-enabled applications in various industries, from smart home devices to virtual assistants and customer support bots. Gaining expertise in speech recognition using Python can open up numerous career opportunities in the tech industry.

Personalization and User Interaction:

Speech recognition allows for the development of personalized applications that can learn and adapt to individual user preferences and speech patterns, providing a more engaging and responsive interaction.

Language and Communication Enhancement:

Projects focused on speech recognition can also contribute to better communication tools, such as real-time translation services, language learning applications, and tools for individuals with speech disorders. By leveraging Python for speech recognition, developers can create innovative.



Fig 3: Speech Recognition

1.3 Existing System:

There are several well-established algorithms and libraries for speech recognition in Python. Here are a few of the most popular ones:

Google Speech Recognition API: Part of the Speech Recognition library, this API provides high accuracy and supports multiple languages. It requires an internet connection to work.

CMU Sphinx (pocket sphinx): An offline speech recognition toolkit that is part of the CMU Sphinx project. It is not as accurate as Google's API but does not require an internet connection.

Mozilla Deep Speech: An open-source speech-to-text engine based on deep learning. It provides a high level of accuracy and can be used both online and offline.

Wit.ai: An API that provides natural language processing capabilities, including speech recognition. It is free to use but requires an internet connection.

1.4 Challenges in Existing System

Challenges in existing system are:

Drawbacks:

Despite the popularity of voice recognition technology, there are some disadvantages to using it. While the automatic speech recognition may be the star of the show, it can still be a bit of a challenge to implement. For example, it may not be able to capture all words accurately because of pronunciation variations, or it may not be able to sort through background noise.

If this is the case, it may be best to rely on a more accurate (and more expensive!) transcription service.

Using automatic speech recognition can be an advantage in some industries, such as the law industry. It can help lawyers reduce time spent on legal research and documenting their cases, and ensure the accuracy of their work. It also allows for more efficient internal processes. However, many users are still hesitant to use an ASR bot for sensitive tasks, for the reason we've mentioned above – lack of accuracy.

There are still many factors to consider before implementing automatic speech recognition in your office. For example, you must ensure that your office equipment is capable of recording quality audio, and that your software can accurately read the text produced.

Although the technology can make documentation easier, it can also result in errors. For instance, the system may not understand accents or slang. It may also take longer than anticipated to capture words correctly.

Some users are worried that they will not be able to trust the voice recognition system. They are hesitant to use ASR bots for sensitive tasks. The lack of trust may cause businesses to hesitate in adopting this technology.

The technology is also expensive to implement. This may include special hardware and software. Depending on the application, it may also require significant training. There may also be regulatory requirements.

Voice recognition software can be a distraction. It may not be able to differentiate between ambient noise and the actual speech. Wearing a noise-cancelling headset may help. Also, people who speak in accents need to learn to speak clearly so the system can recognize them. People must also avoid talking in a choppy manner or mumbling. This can lead to grammar and spelling errors.

Speech recognition may also have data privacy concerns. It is important to consider how a speech recognition system will handle your personal information. This information may include sensitive financial or medical information.

Then, it is often necessary to train your employees in the proper use of automatic speech recognition. This includes developing a training program based on different scenarios. A training program can also include a number of other features, such as filtering out background noise.



Fig 4: Sound Waves

1.5 Proposed System:

Creating a speech recognition project in Python typically involves several steps. Here's a basic methodology to guide you through the process.

1.Setting Up the Environment:

Install necessary libraries: `speech_recognition`, `pyaudio`, and `numpy`.

Ensure you have a working microphone and the necessary permissions to use it.

2. Recording Audio:

Use `pyaudio` or similar libraries to capture audio input from the microphone.

Save the recorded audio as a `.wav` file for processing.

3.Processing Audio:

Use the `speech_recognition` library to convert speech to text.

This involves creating a `Recognizer` object and using it to process the audio.

4.Handling Different Languages and Accents:

Configure the recognizer to handle different languages by specifying the `language` parameter.

Use different recognizer instances or configurations to handle various accents and dialects.

5.Error Handling:

Implement robust error handling to manage issues like background noise, unclear speech, and interruptions.

1.6 Advantages of the Proposed System:

In mostly areas of the country, there are lot of people who don't know how to write and also how to read any word, so this project is very helpful for these type of people as you know in today's world, everybody has its own mobile phones and they want to search a lot of things. In this project, they usually speak what they want to search and various results of such type opens in the browser window.

Ability to write text using speech.

1. Different windows can be opened and web searches can be made.
2. More utilization of resources and less time consumption.
3. Recognizes different audio files and convert them to text.
4. Helpful for disabled peoples.

1.7 Objectives

To be familiar with the speech recognition and its fundamentals.

- Its working and application in different areas.
- To implement it as an application for relative searches.
- Software which can be used for:
 - Speech Recognition
 - Web searches

1.8 Methodology

The basic function of both speech synthesis and speech recognition is easy to understand as there are many powerful capabilities provided by speech recognition technology that helps many developers to understand and utilize this technology.

Despite the substantial growth and research in speech recognition technology there are still more limitations in this technology. Because of the speech recognition humans are able to utilize the time in various aspects and also it proves to be beneficial to various disabled peoples, still this system is unfamiliar with natural human to human conversations. The complete knowledge of the limitation also the strength is very important for the accurate use of speech recognition technologies as there may be differences in the output provided by the system and the output required by the user for a particular input. Due to this understanding the user or developers of these application can make a decisions about whether the technology will benefit the use of speech-to-text in a particular speech input.



Fig 5: Speech Recognition using python

CHAPTER 2

Literature Survey:

HISTORY:

From (1970-2010):

The First speech recognition system was focused on numbers, not words. In 1952 bell Laboratory designed the “Audrey System” which could recognize a single voice speaking digits aloud. Ten years later IBM introduced “shoebox” which understood 16 words in English. Across the globe other nations developed hardware that could recognize sound and sleep. And by the end of ‘60s, the technology could support words with 4 vowels and nine consonants.

From 1970’s:

Speech recognition made several meaningful advancements in this Decade. This was mostly due to the US Department of defence and DARPA. The Speech Understanding Program SUR program there ran was one of the largest of its kind in the history of speech recognition. Mellon ‘Harpy Speech System came from this program and was capable of understanding over 1000 kind words that is about the same a three-year Old’s vocabulary. Also significant in the 70’s was Bell Laboratories introduction odd the system that could interpret multiple voices.

From 1980’s:

The ‘80s saw speech Recognition vocab go from few of hundred’s words to the several thousand words. One of the Breakthroughs that came from a statistical method known as the ‘Hidden Markov Model0 ‘HMM’ ‘. Instead of just using words and looking for the sound patterns. The Hmm estimated the probability of the unknown sounds actually being words.

From 1990's:

Speech recognition was propelled forward in the 90s in the large part because of the own personal computer. The faster processors made it possible for software like dragon dictate to become the more widely used bell south introduced the Voice Portal (VAL) in which was a dial in interactive voice recognition system. This System give new birth to the myriad of the phones tree system that are still in the existence.

From 2000's:

From the year 20002 Speech recognition Technology had achieved close to the 80 percent accuracy. For almost of all the Decade there aren't a lot of Advancements till Google has come with a start of Google search voice. As it was an application which put speech recognition into hands of lakhs of people. This was also Significant because that the processing power would be offloaded to its data Centres. Not only for that, was Google Application collecting data from many billions of the searches which could help this to predict what a human is actually saying. That time Google's English voice search system, included 240 billion words from user searches.

From 2010's:

In 2012 Apple Launched SIRI which was as same as the Google's VOICE SEARCH. The early part of the decade saw an explosion of the other voice Recognition Applications. And with Amazon's ALEXA, Google Home we've seen consumers becoming more and more comfortable talking to Machines. Today, some of the Largest Technical Companies are competing to herald the speech accuracy title. In 2015, IBM achieved a word ERROR RATE of 6.8%. IN 2016 Microsoft overpassed IBM with a 5.8 % claim. Shortly After that IBM improved their Rate to 5.4 %. However, it's Google that claims the lowest Ratio rate at 4.8percent.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Purpose

The primary purpose of a speech recognition system is to *convert spoken language into written text* or to *interpret commands* that the system can process. This technology allows users to interact with devices using their voice, eliminating the need for manual input through keyboards or touchscreens.

Here are the key objectives behind the development and use of speech recognition systems:

1. Automation: Speech recognition systems help automate various tasks like transcription, virtual assistance, voice-controlled commands, and customer service, saving time and effort.

2. Accessibility: It enables users with disabilities, such as those with visual impairments or mobility issues, to interact with technology in an efficient and intuitive way.

3. Efficiency: By allowing hands-free interaction, speech recognition systems improve user efficiency, especially in environments where multitasking or manual input would be cumbersome.

4. Natural User Interface: These systems create a more natural and human-like interface, allowing users to communicate with devices as they would with another person.

5. Improved User Experience: Offering an alternative to typing or tapping, speech recognition can provide a faster, more intuitive way to interact with technology, which is especially valuable in mobile or in-car environments.

3.2 Scope:

The scope of a speech recognition system can vary depending on the application and use case, but generally, it includes the following areas:

1. Voice-to-Text Transcription:

Converts spoken language into written text, which is useful in applications like note-taking, transcription services, and real-time captioning.

Scope includes:

- Dictation applications (e.g., medical transcription).
- Speech-to-text conversion for meetings, lectures, or interviews.

2. Voice Command Recognition:

Recognizes spoken commands and executes corresponding actions.

Scope includes:

- Virtual assistants (e.g., Siri, Alexa, Google Assistant).
- Hands-free control in devices like smartphones, smart speakers, and smart home systems.
- Voice-controlled navigation and other in-vehicle systems.

3. Natural Language Processing (NLP) Integration:

Combines speech recognition with NLP to understand user queries and provide intelligent responses, enabling conversational interfaces.

Scope includes:

- Customer service chatbots or virtual assistants.
- Interactive voice response (IVR) systems.
- Language translation and multi-language support.

4. Security and Authentication:

Voice biometrics can be used to verify the identity of the speaker.

Scope includes:

- Voice-based authentication in banking, security systems, and personal devices.

5. Real-time Speech Translation:

Converts spoken language from one language to another in real-time, enabling cross-lingual communication.

Scope includes:

- Translation apps or services for international communication.
- Simultaneous interpretation for conferences or meetings.

6. Multimodal Interaction:

Integrates speech recognition with other input modalities (e.g., text, gestures) to enable a more flexible and comprehensive user experience.

Scope includes:

- Voice + touch interfaces in smart devices, wearables, and automotive systems.

7. Healthcare Applications:

Used in medical dictation systems, electronic health record (HER) management, and voice-powered medical devices.

Scope includes:

- Physicians dictating patient notes or prescriptions.
- Medical transcription services.

8. Entertainment and Media:

Voice recognition is often used in gaming, media control, and interactive TV.

Scope includes:

- Voice-enabled gaming (e.g., controlling characters or in-game actions).
- Voice-based content navigation in streaming services.

3.3 Feasibility Study:

3.3.1 Economic Feasibility

Economic feasibility examines the financial aspects of the project, including cost analysis, benefits, and potential return on investment (ROI).

1.Development Costs:

- Python is an open-source language, so there are no licensing costs involved. However, the development costs may include hiring skilled developers, paying for infrastructure (e.g., servers, cloud services), and purchasing or developing any proprietary algorithms.
- Some speech recognition libraries and APIs, like Google Cloud Speech-to-Text, IBM Watson, and Microsoft Azure Speech, charge based on usage, which can incur costs. Alternatively, open-source libraries like SpeechRecognition and pyaudio can help reduce costs, but they may require more setup and maintenance.

2.Infrastructure Costs:

- If the system requires extensive computation (e.g., for real-time processing or large-scale deployment), you may need powerful servers or cloud computing services (AWS, Google Cloud, etc.). However, cloud services often offer scalable, pay-as-you-go pricing, which can be cost-effective in the long run.

3.ROI and Benefits:

- Automating tasks like transcription, voice commands, and customer support through speech recognition can significantly reduce operational costs.
- Improved user experience and accessibility can open new markets and attract more customers, potentially leading to higher revenue.
- Time-saving and accuracy improvements can increase employee productivity and reduce manual labor costs.

3.3.2 Technical Feasibility :

Technical feasibility focuses on the ability to implement and support the system from a technology perspective.

Libraries and Frameworks:

Python provides several libraries and tools to build speech recognition systems, making it technically feasible to implement the system. Notable libraries include:

SpeechRecognition: This library allows you to interface with various speech-to-text APIs (e.g., Google Web Speech API, Sphinx, etc.).

pyaudio: Used for capturing audio data from microphones.

Google Speech-to-Text API: A powerful cloud-based tool for accurate transcription.

DeepSpeech: A Mozilla-backed open-source project that uses deep learning for speech recognition.

TensorFlow/Keras/PyTorch: These can be used to build custom deep learning-based models for speech recognition if higher accuracy is needed.

Model Complexity: Basic models may work well for simple use cases, such as command-based systems, but more advanced models are needed for real-time transcription, accent variation, or background noise handling. For high-quality performance, pre-trained deep learning models or custom-trained models (on large datasets) can be leveraged, but this requires technical expertise in machine learning and natural language processing.

Hardware Requirements:

- If using real-time speech recognition, processing speed and low-latency are critical. You might need powerful CPUs, GPUs, or access to cloud-based processing power for fast processing.

- The quality of the microphone and sound system affects the accuracy of speech recognition. High-quality audio inputs can reduce noise and errors, thus improving performance.

Scalability and Maintenance:

- Python-based speech recognition systems can scale using cloud solutions to handle larger volumes of requests. Cloud-based APIs often handle scaling automatically.
- However, maintaining the system, ensuring consistent updates, monitoring accuracy, and improving models over time require ongoing technical effort.

3.3.3 Social Feasibility:

Social feasibility focuses on how the speech recognition system impacts society, users, and the broader community.

1.User Acceptance and Experience:

Accessibility: Speech recognition systems can improve accessibility for individuals with disabilities, such as those with visual or motor impairments. This can lead to better adoption, especially in apps that help people with these challenges.

Language and Accent Support: One challenge in social feasibility is ensuring that the system can support a wide range of languages, accents, and dialects. If the system doesn't accurately recognize diverse speech patterns, it can cause frustration and limit adoption.

Privacy Concerns: Users may be concerned about the privacy of their voice data. Clear data protection policies and providing transparency regarding how voice data is stored and used are important for user trust.

2.Cultural and Ethical Implications:

- Speech recognition systems could inadvertently reflect biases present in training data, such as accent or gender biases, leading to less accurate transcriptions for certain user groups. Ethical considerations in data collection and system design need to be prioritized.

- Speech recognition could be used to enhance customer service (chatbots, virtual assistants) and improve overall user experience. However, if not well-implemented, it could create a feeling of alienation or frustration among users if the system doesn't perform as expected.

Impact on Employment: While speech recognition can improve productivity, it may also displace certain jobs that rely on manual transcription or customer service roles. On the flip side, it can create new roles in AI development, data science, and system maintenance.

3.Regulatory and Legal Compliance:

- In some regions, speech data may be subject to data protection regulations (e.g., GDPR, HIPAA). Ensuring compliance with these regulations is critical to avoid legal consequences and to protect user data.

3.4 REQUIREMENT ANALYSIS:

3.4.1 Functional Requirements:

These are the specific features and functionalities that the system must support:

1. Speech-to-Text Conversion:

- The system must be able to convert spoken words into text in real-time or near real-time.
- Support for various audio formats (e.g., WAV, MP3, etc.) as input.

2. Voice Command Recognition:

- The system should recognize predefined voice commands and perform corresponding actions (e.g., "open file", "play music").
- The ability to handle simple or complex commands, depending on the use case.

3. Support for Multiple Languages and Accents:

- The system should support multiple languages (e.g., English, Spanish, French) and regional accents.
- The ability to detect the language automatically or allow the user to select their language preference.

4.Real-Time Processing:

- The system must provide real-time or near-real-time transcription for continuous speech input.
- It should handle live voice inputs, such as from a microphone, and immediately convert speech to text.

5. Noise Handling:

- The system should have the ability to filter out background noise or adjust for various environmental conditions (e.g., noisy environments, varying microphone quality).

6. Error Detection and Correction:

- The system must identify and correct errors or inaccuracies in transcription, with options for manual correction or automatic error handling.

7. Voice Command Feedback:

- The system should provide audio or visual feedback to the user when a command is successfully recognized and executed (e.g., confirmation message, action).

8. Integration with External APIs:

- The system should integrate with external speech recognition APIs (e.g., Google Speech API, IBM Watson) or open-source libraries (e.g., SpeechRecognition, PyAudio) for enhanced accuracy and functionality.

3.4.2 Non-Functional Requirements:

These are the qualities and constraints of the system that ensure it operates effectively in terms of performance, reliability, and usability:

1. Performance:

- Speed: The system should provide near-instantaneous or real-time transcription with minimal delay.
- Accuracy: The system should transcribe speech with a high degree of accuracy, ideally above 90% in ideal conditions.
- Scalability: The system should be capable of handling an increasing number of requests or larger datasets without significant performance degradation.

2. Usability:

- Ease of Use: The system should be user-friendly, with a simple interface for initiating voice input and interacting with the transcriptions or commands.
- Clear Instructions and Feedback: The system should provide clear instructions to the user and feedback when processing voice commands, including success or failure messages.

3. Compatibility:

- The system should work across various platforms (e.g., Windows, macOS, Linux) with minimal adjustments.
- It should be compatible with common audio input devices (e.g., microphones, headsets).
- Cross-browser support (if it's a web-based application) or support for desktop/mobile environments (if standalone).

4. Security:

- The system should ensure the confidentiality and security of the recorded audio and transcribed data.
- The system should provide user authentication (if required) and encryption for sensitive data.
- Compliance with privacy regulations such as *GDPR* (General Data Protection Regulation) should be considered when handling voice data.

5. Reliability:

- The system should operate with minimal downtime, offering high availability for users.
- It should handle errors gracefully and provide meaningful error messages in case of failure (e.g., "Unable to recognize speech", "Microphone not found").

6. Maintainability:

- The system should be easy to maintain and update, including adding new voice commands, languages, or improving the speech recognition model.
- Clear and structured documentation for developers and users should be available.

7. Robustness:

- The system should handle different voice inputs, including different speaking speeds, accents, and clarity of speech.
- It should also handle interruptions, such as pauses in speech, background noise, and overlapping voices, gracefully.

8. Accessibility:

- The system should be designed to be accessible to people with disabilities (e.g., visually impaired users).
- Voice feedback should be available, and the system should be usable without requiring manual inputs.

3.5 REQUIREMENT SPECIFICATION:

3.5.1 Hardware Requirements:

Microphones:

Microphones are the most important tools for the real time speech to text conversion in Therefore the pre-installed ones cannot be used as they are more prone to the background noise and also of poor quality terms of speech.

Computer Processor:

Speech recognition application depends majorly on processing speed. The input from the user can take some time if the processing speed is low and thus user wasted more time on waiting compared to performing the task which makes the application less feasible for use .

3.5.2 Software Requirements:

1.6 MHz Processor

128 MB RAM

Microphones for good audio.

Best Requirements:

- a. 2.4 GHz processor
- b. Greater than 128 MB RAM
- c. 10% consumption of memory
- d. Best quality microphones

3.5.3 Language Specification:

Primary Language: Python 3.x.

Core Libraries: SpeechRecognition, PyAudio, optional integrations with APIs like Google Speech or IBM Watson.

Supporting Libraries: Libraries like NLTK or PyDub can enhance functionality depending on your needs (e.g., text analysis or audio preprocessing).

Code Organization: Well-structured project folders with clear file naming conventions and modular code. This structure and set of specifications help ensure your Python-based speech recognition system is efficient, maintainable, and scalable.

CHAPTER 4

SYSTEM DEVELOPMENT DESIGN

4.1 Speech synthesis :

4.1.1 Evaluation of synthetic speech:

Speech Synthesis Systems can be calculate I terms of different requirements such as speech intelligibility, Speech Naturalness, System Complexity, and so on. For Ambient Intelligent Application it is Reasonable to imagine that new Evaluation Criteria will be Require for example , emotional Influence on the User, Ability to get the User to Act, mastery over Language generation, and Whether the system takes the Environmental Variables into Account and adjusts its behaviour Accordingly.

Some Of the Just Mentioned evaluation Criteria are for the Complete System. Having Evaluation Criteria for the Whole System is reasonable because a single, miss performing component would negatively impact how the system is perceived by humans.

4.1.2 Building Speech Synthesis Systems:

Building Speech Synthesis Systems require a speech Units Corpus. Natural Speech must have been recorded for all Units- For Example, all Phonemes – in all possible Contexts.

Next the Units in the Spoken Speech Data are segmented and labelled. Finally, the most Appropriate Speech Units are Chosen (Black and Campbell, 1995).

Generally, concatenative Synthesis yields high quality Speech. With the Large Speech Units Corpus, high quality speech waveforms can be generated. Such synthesized speech preserves waveforms can be generated. Such synthesized speech preserves naturalness and intelligibility. Separate prosody modelling is not necessary for speech unit selection due to the availability of many units corresponding to varied contexts. Picture

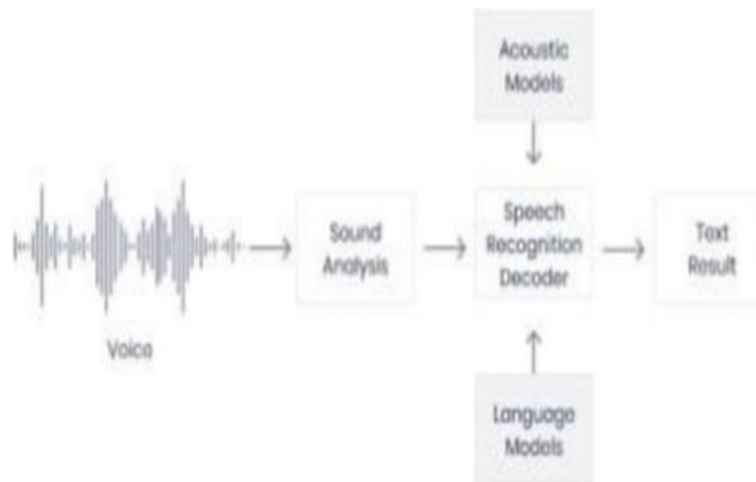


Fig 6: System Development

4.2 Packages used :

The following are the packages installed in this project:

1.Import speech recognition:

Speech recognition helps to take the input with ease and helps in running model in just a few minutes. The speech recognition library has several popular speech APIs and is thus extremely flexible. It consists of seven APIs which can be used to speech recognition but all six APIs comes with authentication key and password except Google speech API which makes it extremely flexible and with its ability of free usage and ease of use it makes it excellent choice for speech recognition.

2.Import audio:

The pip install Audio command installs py audio to the python interpreter and thus make it easier to work with microphones which helps in real time speech recognition. With Py Audio, we can easily use Python to record and to play audio on a kind of variety of platforms.

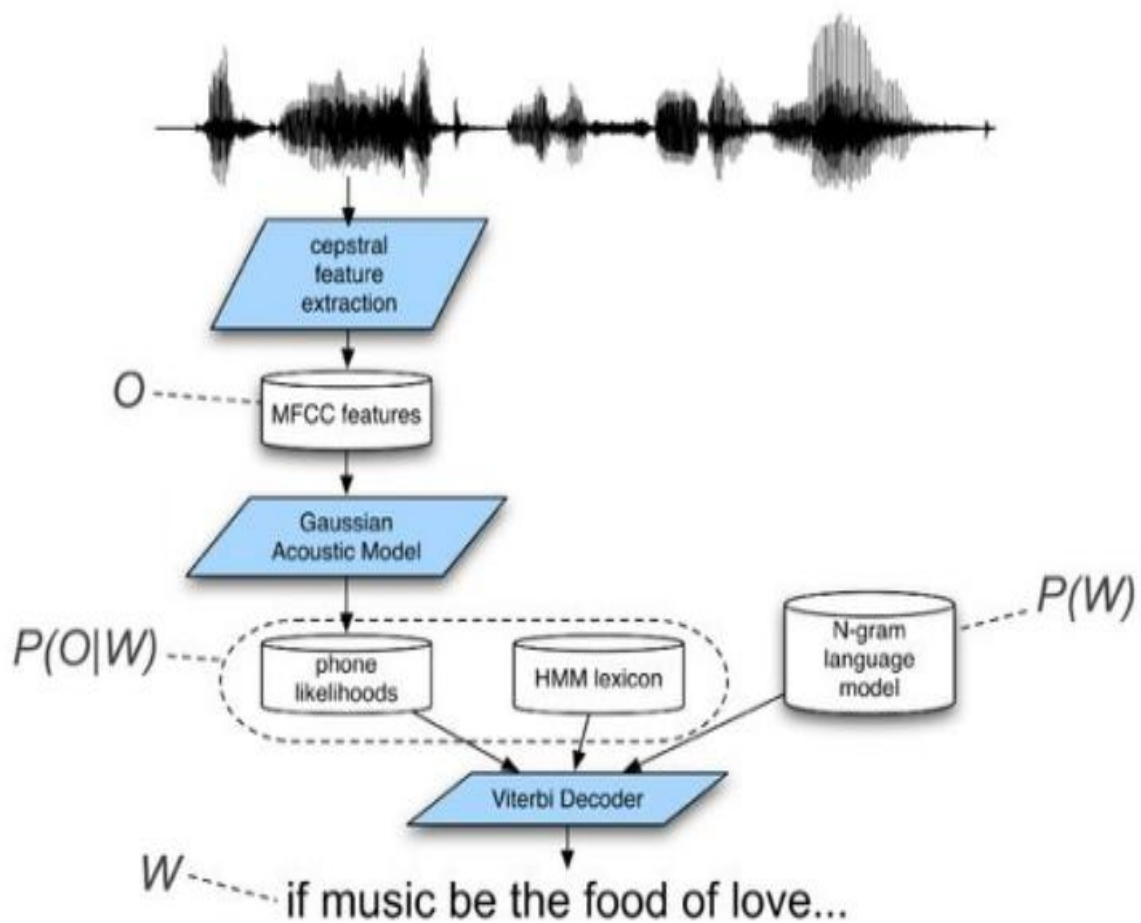
3.Import web browser:

With this package we can make use of our default browser used to locate, retrieve and display data .The URL and the query is passed to the instance of the web browser package and basis on the" URL" provided and the query the particular webpage opens. Speech Recognition is an important feature in several applications used such as home automation, artificial intelligence, etc. This article aims to provide an introduction on how to make use of the Speech Recognition and pyttsx3 library of python.

4.3 System architecture :

The architecture includes an audio capture component, where spoken input is captured through a microphone. Analog-to-digital conversion translates the captured audio for processing. Acoustic modeling converts audio into phonetic units, followed by language modeling to interpret meaning. Display and response functions show the processed text and execute corresponding actions.

Speech Recognition Architecture



4.4 UML diagrams :

Some of the frequently used use case diagrams in software development are:

- Use Case Diagram
- Activity Diagram
- Sequence Diagram
- Class Diagram

4.4.1 Use Case Diagram:

A use case diagram shows how users interact with a system by representing actors, use cases, and their relationships. It is useful for understanding system functionality and defining requirements clearly.

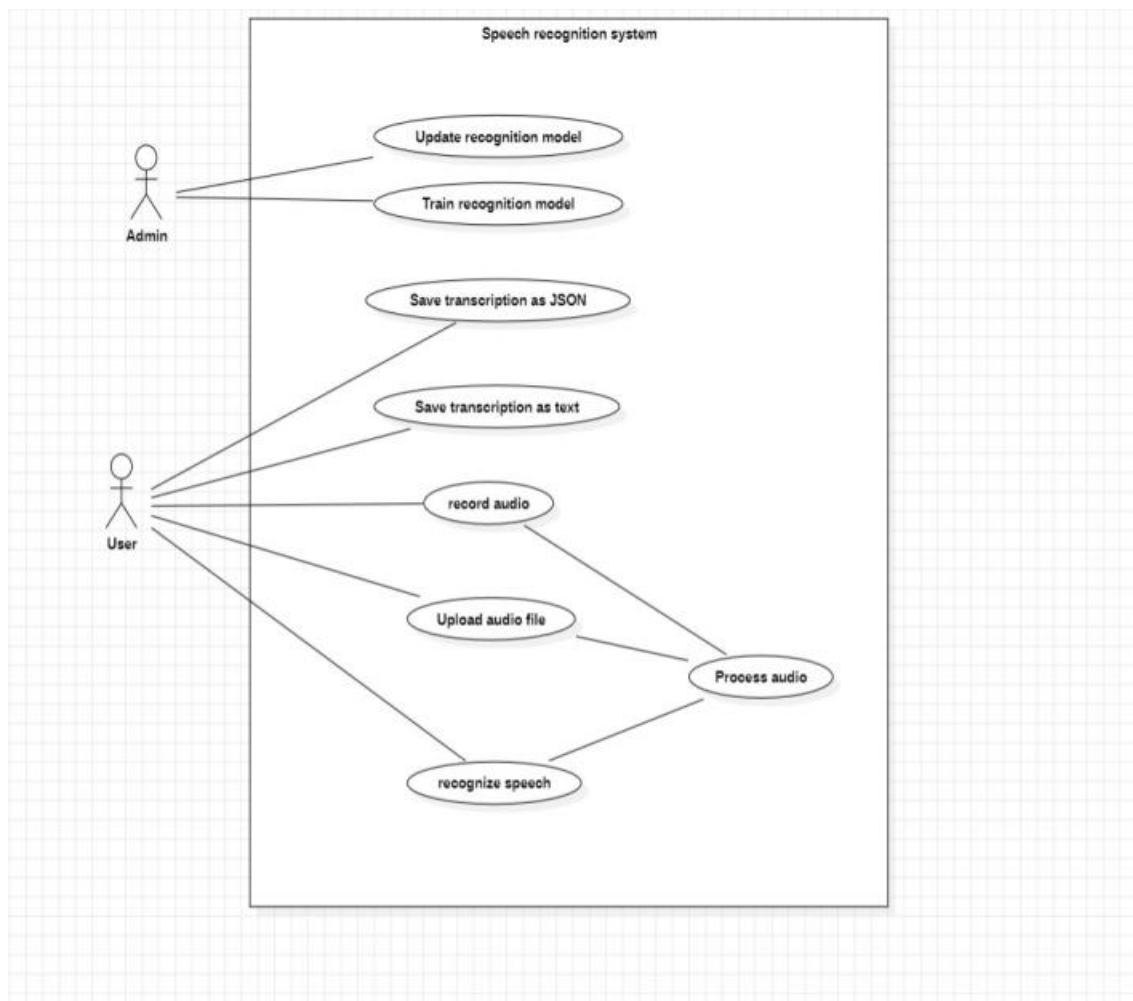


Fig: Use Case Diagram

4.4.2 Activity Diagram:

An activity diagram represents the flow of activities or actions in a system, showing how tasks are performed and how they are connected. It highlights the sequence and conditions of processes, making it useful for understanding and improving workflows or processes in a system.

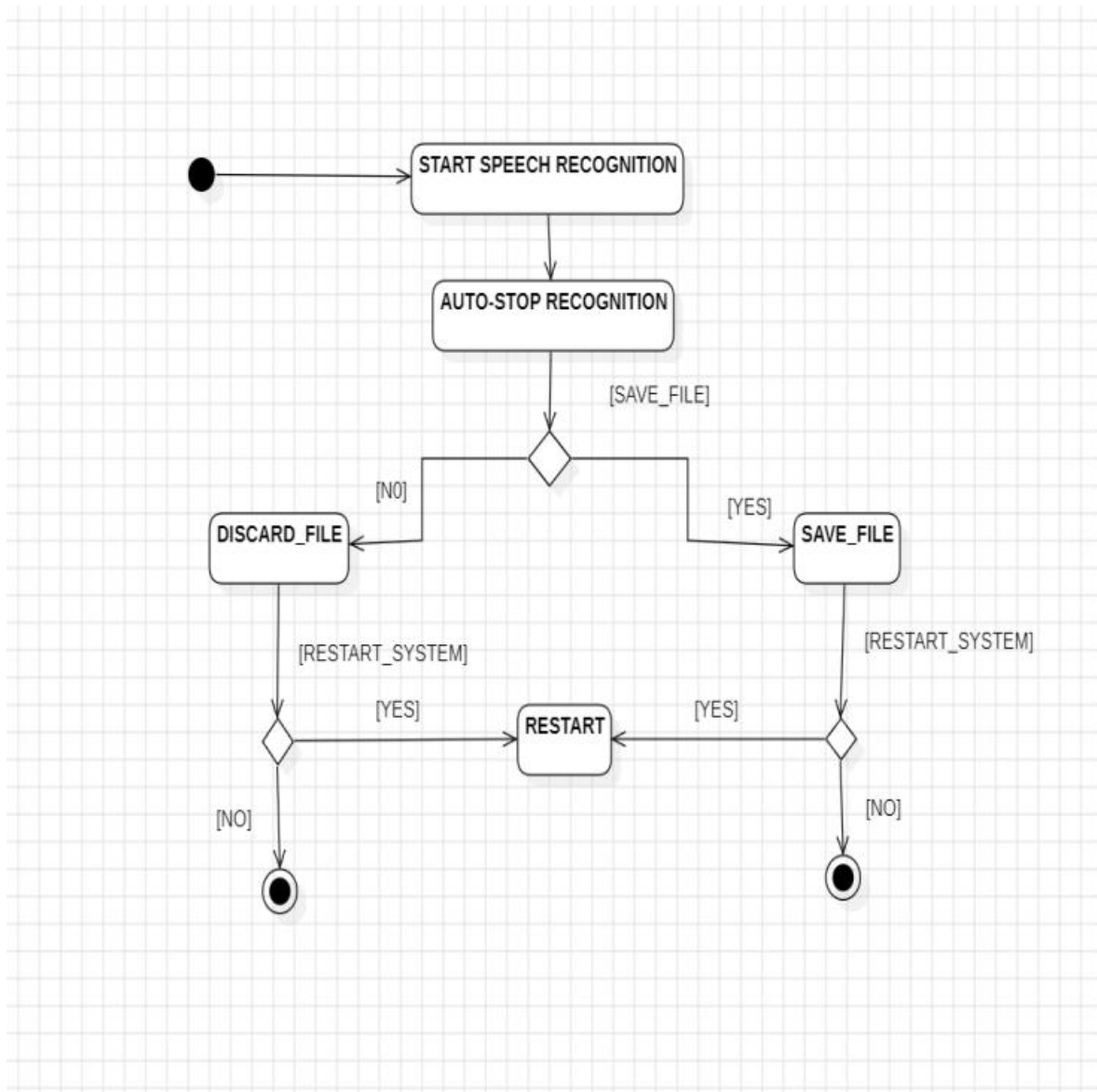


Fig:Activity Diagram

4.4.3 Sequence Diagram:

A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. Sequence diagram used lifeline which is a named element which depicts an individual participant in a sequence diagram. Communications happens as the messages appear in a sequential order on the lifeline.

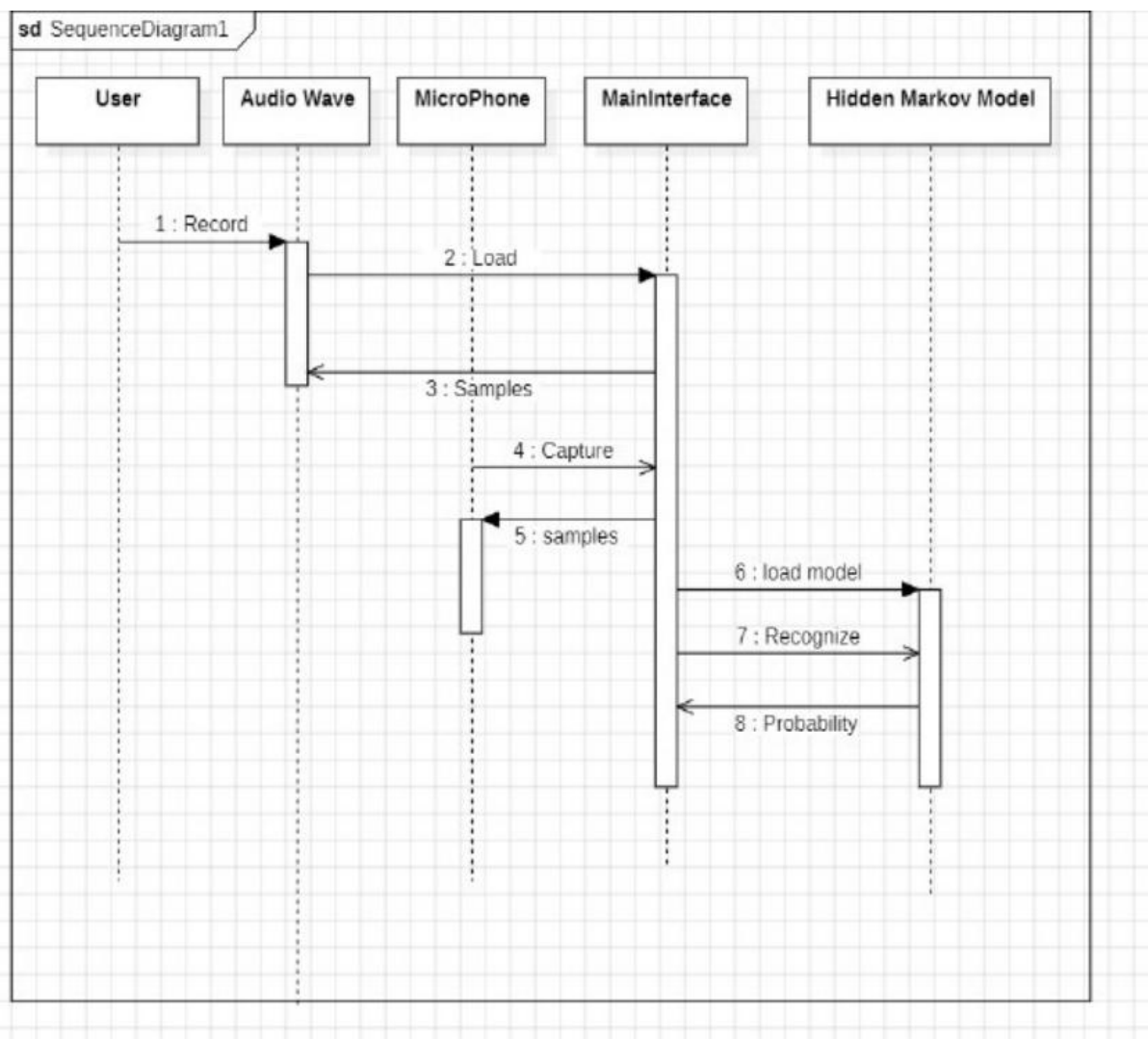


Fig: Sequence Diagram

4.4.4 Class Diagram:

The class diagram depicts a static view of an application. It represents the types of Objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes.

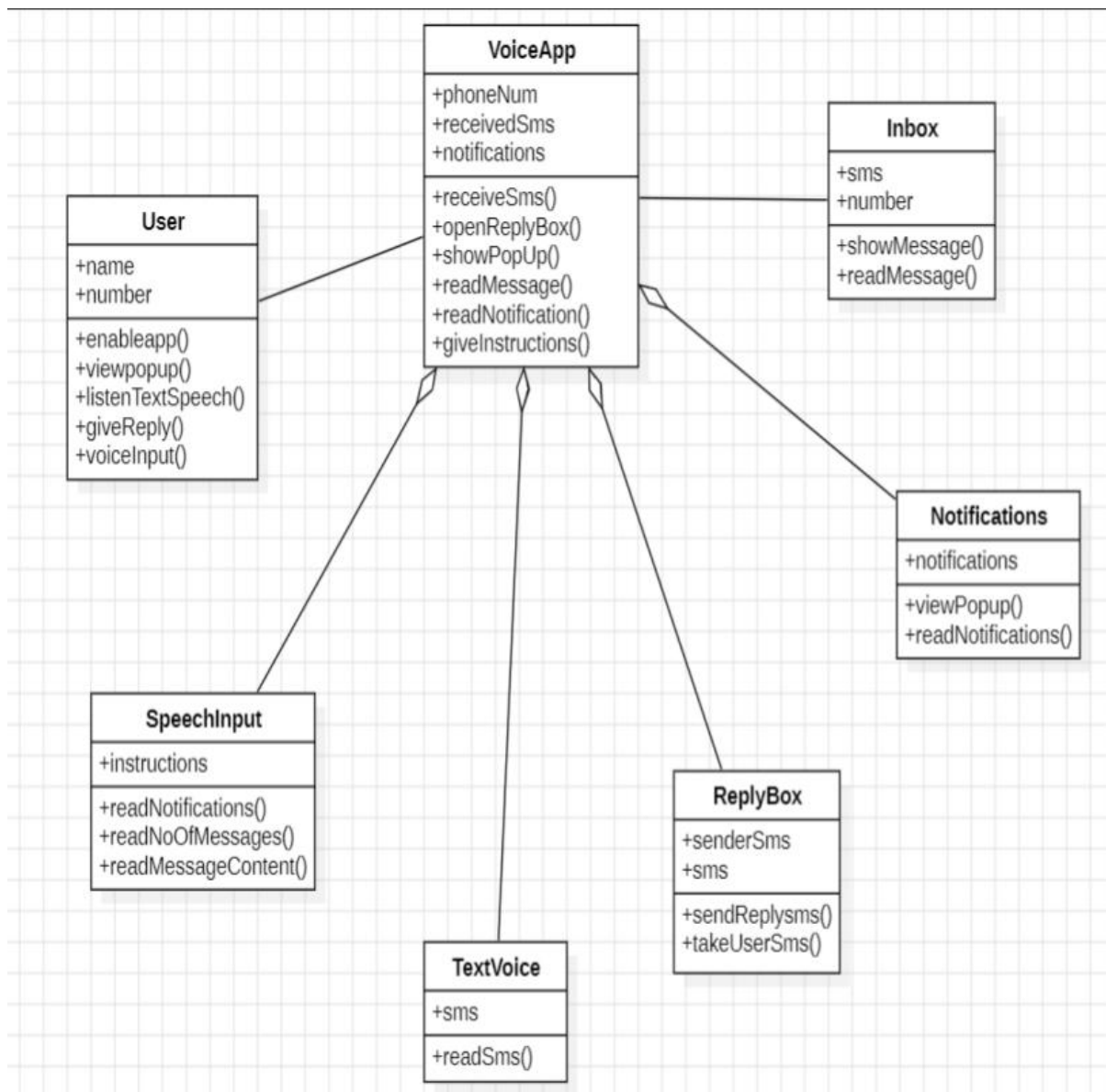


Fig: Class Diagram

CHAPTER 5

IMPLEMENTATION AND RESULTS:

5.1 Methods or Algorithms used:

1. Speech Recognition Library:

- Utilized speech_recognition Python library for processing speech and converting it into text.
- Several APIs were supported (e.g., Google Web Speech API, IBM Speech to Text), with recognize_google() highlighted for use.

2. Audio File Handling:

- Implemented the AudioFile class for handling and processing audio data.
- Used methods like record() for extracting audio and parameters like offset and duration to modify processing regions.

3. Noise Adjustment:

- Employed the adjust_for_noise method to minimize the impact of environmental noise on recognition accuracy.

4. Microphone Input:

- Used PyAudio to access real-time speech input from a microphone.

5. Guess-the-Word Game:

- Implemented a speech-recognition-based game where the user guesses words using a microphone input.

6. Web Browsing Integration:

- Speech input converted into commands to perform web searches and query-specific searches using the webbrowser package.

5.2 Sample Code:

```
# Step 1: Install necessary libraries
!pip install SpeechRecognition pydub

from google.colab import files
import speech_recognition as sr
from pydub import AudioSegment

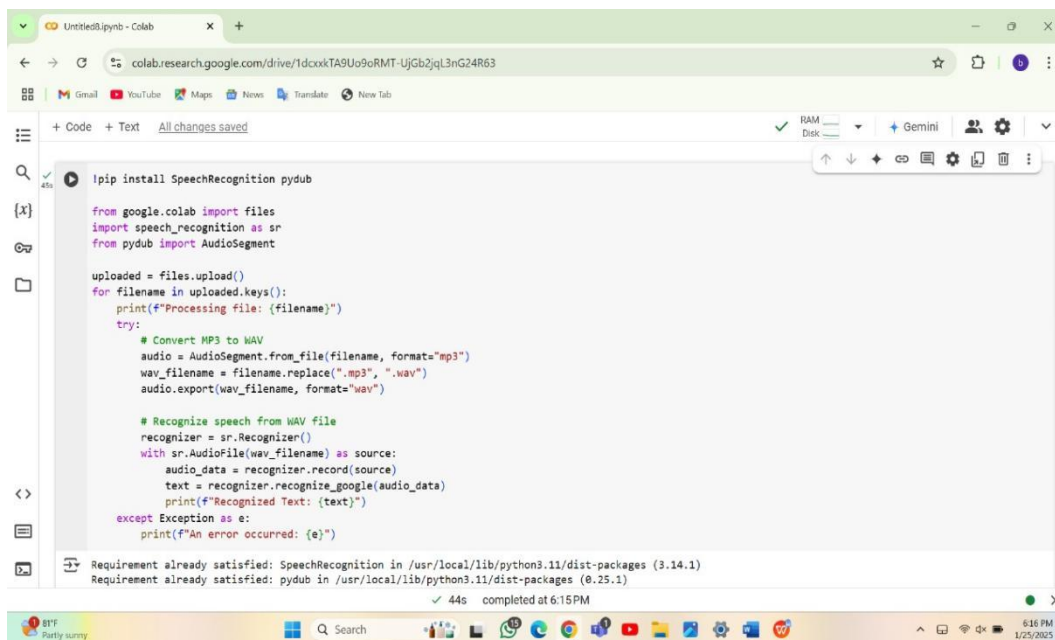
# Step 2: Upload an MP3 audio file
uploaded = files.upload()

# Step 3: Convert MP3 to WAV and recognize speech
for filename in uploaded.keys():
    print(f'Processing file: {filename}')
    try:
        # Convert MP3 to WAV
        audio = AudioSegment.from_file(filename, format="mp3")
        wav_filename = filename.replace(".mp3", ".wav")
        audio.export(wav_filename, format="wav")

        # Recognize speech from WAV file
        recognizer = sr.Recognizer()
        with sr.AudioFile(wav_filename) as source:
            audio_data = recognizer.record(source)
            text = recognizer.recognize_google(audio_data)
            print(f'Recognized Text: {text}')
    except Exception as e:
        print(f'An error occurred: {e}')
```

CHAPTER 6

SCREENSHOTS:



The screenshot displays a Google Colab notebook interface. The top bar shows the notebook title 'Untitled3.ipynb - Colab' and the URL 'colab.research.google.com/drive/1dcxkTASUo9oRMT-UjGb2jql3nG24R63'. The left sidebar contains icons for file explorer, search, and other notebook functions. The main code editor area contains the following Python code:

```
!pip install SpeechRecognition pydub

from google.colab import files
import speech_recognition as sr
from pydub import AudioSegment

uploaded = files.upload()
for filename in uploaded.keys():
    print(f"Processing file: {filename}")
    try:
        # Convert MP3 to WAV
        audio = AudioSegment.from_file(filename, format="mp3")
        wav_filename = filename.replace(".mp3", ".wav")
        audio.export(wav_filename, format="wav")

        # Recognize speech from WAV file
        recognizer = sr.Recognizer()
        with sr.AudioFile(wav_filename) as source:
            audio_data = recognizer.record(source)
            text = recognizer.recognize_google(audio_data)
            print(f"Recognized Text: {text}")
    except Exception as e:
        print(f"An error occurred: {e}")
```

The bottom of the notebook shows the output of the code execution, indicating that the required packages are already satisfied:

```
Requirement already satisfied: SpeechRecognition in /usr/local/lib/python3.11/dist-packages (3.14.1)
Requirement already satisfied: pydub in /usr/local/lib/python3.11/dist-packages (0.25.1)
```

The status bar at the bottom of the notebook shows '44s completed at 6:15 PM'.

Colab interface showing a Jupyter Notebook with Python code for audio processing and speech recognition.

Browser address: colab.research.google.com/drive/1dcxkTA9Uo9oRMT-UjGb2jql3nG24R63

Code cell content:

```
uploaded = files.upload()
for filename in uploaded.keys():
    print(f"Processing file: {filename}")
    try:
        # Convert MP3 to WAV
        audio = AudioSegment.from_file(filename, format="mp3")
        wav_filename = filename.replace(".mp3", ".wav")
        audio.export(wav_filename, format="wav")

        # Recognize speech from WAV file
        recognizer = sr.Recognizer()
        with sr.AudioFile(wav_filename) as source:
            audio_data = recognizer.record(source)
            text = recognizer.recognize_google(audio_data)
            print(f"Recognized Text: {text}")
    except Exception as e:
        print(f"An error occurred: {e}")
```

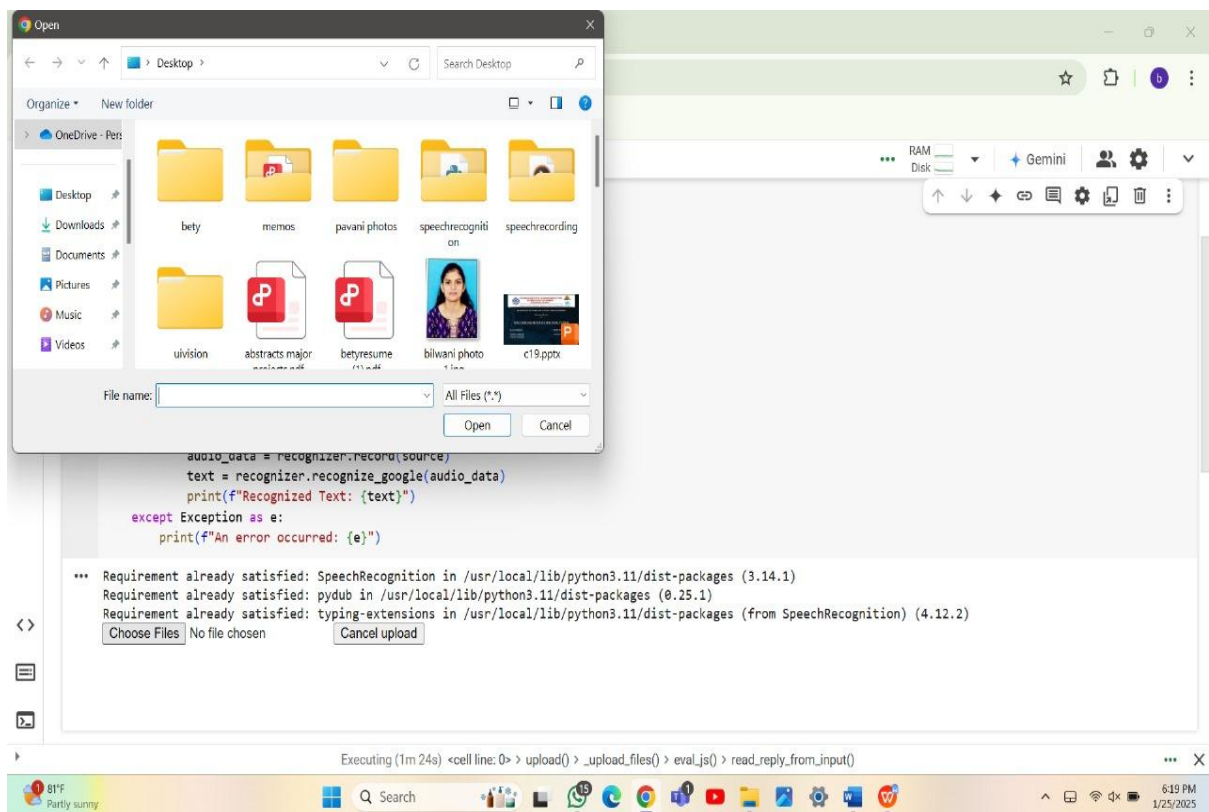
Output:

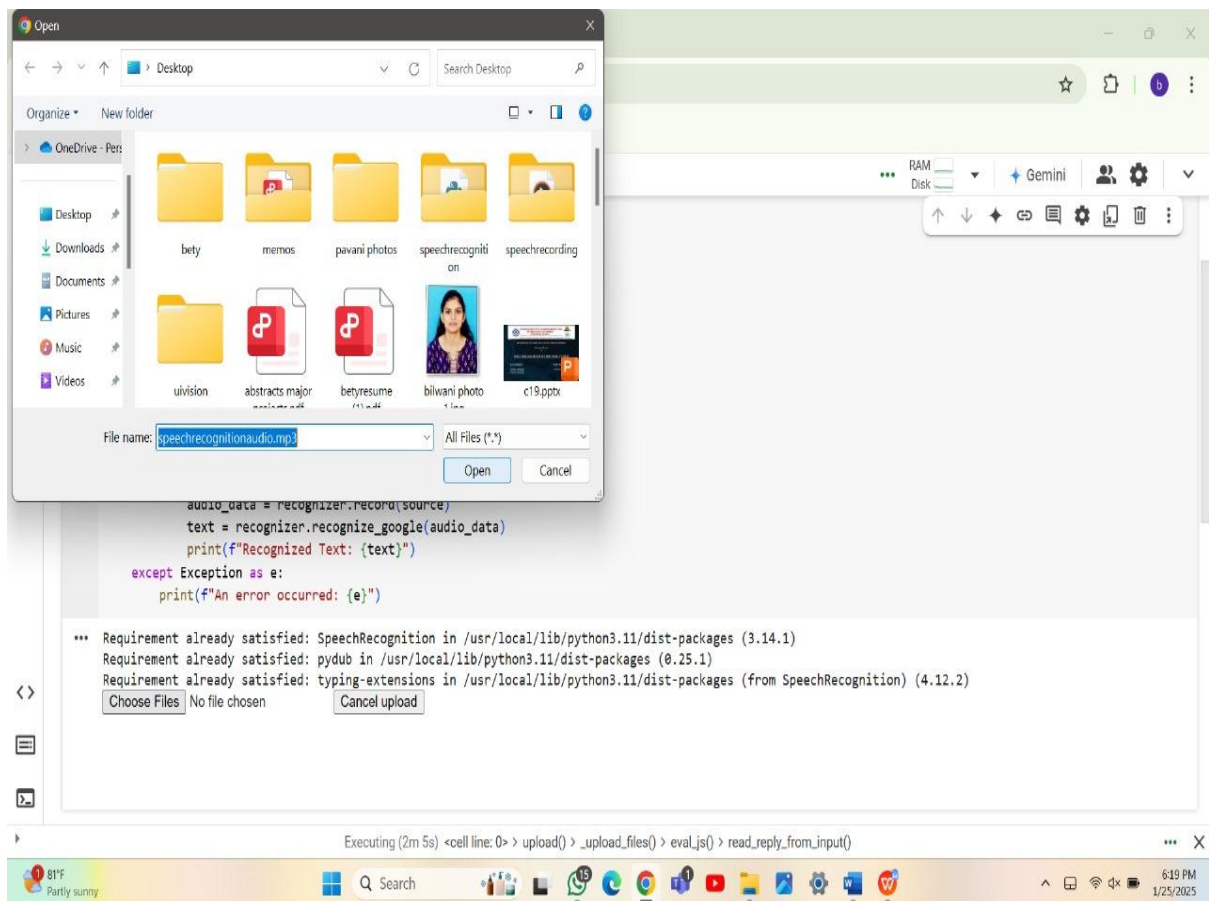
```
*** Requirement already satisfied: SpeechRecognition in /usr/local/lib/python3.11/dist-packages (3.14.1)
Requirement already satisfied: pydub in /usr/local/lib/python3.11/dist-packages (0.25.1)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.11/dist-packages (from SpeechRecognition) (4.12.2)
```

Buttons: Choose Files, No file chosen, Cancel upload

Execution status: Executing (29s) <cell line: 0> > upload() > _upload_files() > eval_js() > read_reply_from_input()

System tray: 81°F Partly sunny, Search, 6:18 PM 1/25/2025





Colab interface showing a Jupyter Notebook titled "Untitled8.ipynb - Colab". The browser address bar shows the URL: colab.research.google.com/drive/1dxxxTA9Uo9oRMT-UjGb2qL3nG24R63.

The notebook contains the following Python code:

```
# Convert MP3s to WAV
audio = AudioSegment.from_file(filename, format="mp3")
wav_filename = filename.replace(".mp3", ".wav")
audio.export(wav_filename, format="wav")

# Recognize speech from WAV file
recognizer = sr.Recognizer()
with sr.AudioFile(wav_filename) as source:
    audio_data = recognizer.record(source)
    text = recognizer.recognize_google(audio_data)
    print(f"Recognized Text: {text}")
except Exception as e:
    print(f"An error occurred: {e}")
```

The output of the code execution is displayed below the code cell:

```
Requirement already satisfied: SpeechRecognition in /usr/local/lib/python3.11/dist-packages (3.14.1)
Requirement already satisfied: pydub in /usr/local/lib/python3.11/dist-packages (0.25.1)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.11/dist-packages (from SpeechRecognition) (4.12.2)
Choose Files speechreco...naudio.mp3
• speechrecognitionaudio.mp3(audio/mpeg) - 238080 bytes, last modified: 1/25/2025 - 100% done
Saving speechrecognitionaudio.mp3 to speechrecognitionaudio (2).mp3
Processing file: speechrecognitionaudio (2).mp3
Recognized Text: speech recognition system using python helps us to convert speech into text
```

The bottom status bar indicates the execution completed at 6:20 PM on 1/25/2025.

CHAPTER 7

SYSTEM TESTING:

1. Testing Modules:

- Each module, such as speech-to-text conversion and web browsing commands, was tested individually.
- Noisy audio files were tested to assess the impact of background noise.

2. Performance Metrics:

- System accuracy was assessed by comparing recognized text against expected output.
- Considered factors like accent, speed of speech, and vocabulary size.

3. Hardware Requirements:

- Ensured testing was conducted on hardware meeting both minimum and best requirements for optimized performance.

4. Practical Use Cases:

- Tested scenarios like web searches, transcription of audio files, and interactive games.

CHAPTER 8

CONCLUSION:

Speech recognition systems built with Python provide a significant advancement in human-computer interaction. Using libraries like SpeechRecognition, pyaudio, and machine learning models, we can convert spoken language into text with a high degree of accuracy. These systems have proven effective in real-world applications such as voice assistants (like Siri and Alexa), transcription services, and accessibility tools for individuals with disabilities. The development of these systems with Python is advantageous due to its rich ecosystem of libraries, ease of use, and strong community support. The performance of the system is dependent on various factors such as background noise, accent variations, and the quality of the microphone.

The implementation of a speech recognition system using Python demonstrates the practical application of natural language processing and machine learning technologies. By leveraging Python libraries such as SpeechRecognition, pyaudio, and Google Speech API, this project successfully translates spoken language into text, providing a foundation for various real-world applications.

This project showcases the potential of speech recognition in streamlining tasks like voice commands, transcription, and accessibility solutions for individuals with disabilities. While the system performs efficiently under controlled conditions, it also highlights areas for improvement, such as handling accents, background noise, and real-time processing speed.

This project not only highlights the capabilities of Python in creating innovative solutions but also opens doors for further exploration and enhancements, such as integrating advanced machine learning models, improving accuracy in noisy environments, and supporting multiple languages. This marks a significant step toward making human-computer interaction more natural and seamless.

CHAPTER 9

FUTURE SCOPE:

Improved Accuracy with Deep Learning:

- Incorporating advanced deep learning models such as deep neural networks (DNN), recurrent neural networks (RNN), and transformers like BERT can significantly improve speech recognition accuracy, especially in noisy environments or with diverse accents.

Multilingual Recognition:

- Developing systems that can recognize and transcribe speech in multiple languages seamlessly is a major area for growth. The system could be trained on multilingual datasets to handle diverse linguistic inputs.

Contextual Understanding:

- Future systems could incorporate natural language processing (NLP) for better contextual understanding, allowing them to not just transcribe speech but also comprehend the meaning behind the spoken words and generate more appropriate responses.

Integration with IoT Devices:

- Speech recognition systems can be integrated into Internet of Things (IoT) devices, allowing users to control smart devices (such as lights, thermostats, and home security) using only their voice.

Real-time Transcription and Translation:

- Implementing real-time speech-to-text and automatic language translation could lead to seamless communication between people speaking different languages, further breaking down language barriers in global communication.

Robustness to Accents and Noises:

- Future research and development in speech recognition will focus on making these systems more robust to variations in accents, dialects, and background noise, thus improving usability in a wider range of environments.

CHAPTER 10

BIBLIOGRAPHY

10.1 References and Websites

[1] KOGILA RAGHU Speech emotion recognition system performance analysis with optimized features using different classification algorithm

[2]

<https://www.ijert.org/python-powered-speech-to-text-a-comprehensive-survey-and-performance-analysis>

[3]

https://www.researchgate.net/publication/343934770_Speech_Recognition_System_A_review

[4]

<http://www.ir.juit.ac.in:8080/jspui/bitstream/123456789/6516/1/Speech%20Recognition%20Using%20Python.pdf>