**A PROJECT REPORT**

***On***

# VOICE RECOGNITION SYSTEM

***in***
***partial fulfillment of the course***
***of***

## PG DIPLOMA

***In***

## BIG DATA ANALYTICS

***From***

## Centre of Development of Advance Computing (CDAC)
## Bangalore, Karnataka

**Guided by:**
**Mr. Jitendra Kumar**

**Presented By**


| | |
|---|---|
| Shriraj Mahadev Gaikwad | PRN: 230950125065 |
| Prayash Raj Kushwaha | PRN: 230950125046 |
| Rishap Kumar | PRN: 230950125052 |
| Amit | PRN: 230950125008 |

# Centre of Development of Advance Computing (CDAC)
# Bangalore, Karnataka

---

## CERTIFICATE

This certificate acknowledges the successful completion of the project on Voice Recognition System by the following individuals:

Shriraj Mahadev Gaikwad (230950125065)
Prayash Raj Kushwaha (230950125046)
Rishap Kumar (230950125052)
Amit (230950125008)

This project was carried out under the guidance of Mr. Jitendra Kumar. Their dedication and efforts in completing the project are truly commendable.

**Forwarded by**

**Mr. Jitendra Kumar**
*Project Guide*
ACTS, CDAC, Bangalore, Karnataka

**Aishwarya Pandey**
*Course Coordinator*
ACTS, CDAC, Bangalore, Karnataka

**Ms. Priyanka Sinha**
*Project Coordinator*
ACTS, CDAC, Bangalore, Karnataka

# Centre of Development of Advance Computing (CDAC)
# Bangalore, Karnataka

---

## DECLARATION

We, the undersigned, hereby declare that the project on Voice recognition titled "Voice Recognition System" has been successfully completed by us, and the work presented in this project report is our own. The individuals involved in this project are:

Shriraj Mahadev Gaikwad (230950125065)
Prayash Raj Kushwaha (230950125046)
Rishap Kumar (230950125052)
Amit (230950125008)

We further declare that to the best of our knowledge, this work has not been submitted, either in part or in whole, for any other academic purpose at this or any other institution.

Shriraj Mahadev Gaikwad
Prayash Raj Kushwaha
Rishap Kumar
Amit

# Centre of Development of Advance Computing (CDAC)
# Bangalore, Karnataka

---

## ACKNOWLEDGEMENT

This project “Voice Recognition System” was a great learning experience for us and we are submitting this work to Advanced Computing Training School (CDAC ACTS), Bengaluru.

We all are very glad to mention the name of **Mr. Jitendra Kumar** for his valuable guidance to work on this project. His guidance and support helped us to overcome various obstacles and intricacies during the course of project work.

Our most heartfelt thanks goes to Ms. Aishwarya Pandey (Course Coordinator, PG-DBDA) who gave all the required support and kind coordination to provide all the necessities like extra Lab hours to complete the project and throughout the course up to the last day at C-DAC ACTS, Bangaluru.

| | |
|---|---|
| Shriraj Mahadev Gaikwad | 230950125065 |
| Prayash Raj Kushwaha | 230950125046 |
| Rishap Kumar | 230950125052 |
| Amit | 230950125008 |

# TABLE OF CONTENT

# ABSTRACT

Voice recognition technology has become a standard feature in many consumer devices, including smartphones, smart speakers, and wearables. The primary objective of this project is to design and implement a Python-based End-to-End voice-to-text recognition system. The dataset used in this project is the "Mini Speech Commands" dataset for training and testing, featuring essential vocal commands like 'yes,' 'no,' 'go,' 'down,' 'left,' 'right,' 'stop,' and 'up.' The dataset is subjected to a meticulous train-test split to ensure a robust evaluation of model performance. The audio files within the dataset undergo a transformation into spectrograms, facilitating the conversion from analog signals to digital representations using Short-Term Fourier Transform (STFT). This preprocessing step enhances the model's ability to extract relevant features from the audio data. Subsequently, a Convolutional Neural Network (CNN) model is trained on the preprocessed dataset, incorporating the spectrogram representations. The CNN model aims to capture intricate patterns and relationships within the spectral domain, optimizing the recognition of distinct voice commands. The evaluation of the model's performance involves assessing its accuracy, precision, and recall on the test dataset. The results provide valuable insights into the model's efficacy in accurately transcribing various speech commands. To make the system accessible and user-friendly, the trained model is deployed using the Gradio library in Python, enabling the creation of an interactive interface. This deployment enhances the project's usability, allowing users to interact seamlessly with the voice recognition system. The project successfully combines the ease of user interaction through Gradio with the accurate speech-to-text capabilities powered by Whisper, leveraging the Mini Speech Commands dataset for diverse and effective training.

## List of figures

| Figure No. | Figure name |
|---|---|
| 1 | Model Architecture |
| 2 | System design of the complete process |
| 3 | Model UI |
| 4 | Graphical Interface |

## List of Abbreviations

| Abbreviations | Meaning |
|---|---|
| CNN | Convolutional Neural Network |
| STFT | Short-Term Fourier Transform |
| IDE | Integrated Development Environment |
| GPU | Graphics Processing Unit |
| CLI | Command Line Interface |
| OS | Operating System |
| UI | User Interface |
| NLP | Natural Language Processing |

# 1. INTRODUCTION AND OVERVIEW

Voice recognition technology has emerged as a ubiquitous feature in contemporary consumer electronics, permeating devices such as smartphones, smart speakers, and wearables. This technology leverages advancements in machine learning and signal processing to convert spoken words into text, enabling seamless human-machine interaction. The increasing integration of voice recognition in our daily lives underscores its significance in enhancing user experience and accessibility.

## Objectives of this project

The primary focus of this project is to design and implement an End-to-End voice-to-text recognition system using Python. The system utilizes a Convolutional Neural Network (CNN) model trained on the "Mini Speech Commands" dataset, a collection encompassing fundamental vocal commands like 'yes,' 'no,' 'go,' 'down,' 'left,' 'right,' 'stop,' and 'up.' This project aims to provide an in-depth exploration of the entire pipeline, from dataset preparation to model training and deployment, with a keen emphasis on optimizing performance and user interaction.

- The project begins with a meticulous examination of the "Mini Speech Commands" dataset, ensuring its suitability for training and testing. The dataset undergoes a careful train-test split to foster a robust evaluation of the model's capabilities.
- To enhance the model's ability to discern relevant features from audio data, the audio files are transformed into spectrograms using the Short-Term Fourier Transform (STFT). This preprocessing step serves as a crucial foundation for the subsequent stages of the project.
- The core of the project revolves around the implementation of a Convolutional Neural Network (CNN) model. This model is specifically tailored to process the spectrogram representations, capturing intricate patterns and relationships within the spectral domain. The objective is to optimize the recognition of distinct voice commands, thereby ensuring the system's accuracy in transcribing spoken words.
- The evaluation phase involves a comprehensive assessment of the model's performance on the test dataset. Key metrics such as accuracy, precision, and recall are employed to gauge the model's efficacy in accurately transcribing various speech commands. These results provide valuable insights into the system's reliability and its potential applications in real-world scenarios.

**Applications of Speech Recognition**

There are more tools accessible for operating this technological breakthrough because it is mostly a software creation that does not belong to anyone company. Because of this, even developers with little financial resources have been able to use this technology to create innovative apps. The following are some of the sectors in which voice recognition is gaining traction

- ***Evolution in search engines:*** Speech recognition will aid in improving search accuracy by bridging the gap between verbal and textual communication.
- ***Impact on the healthcare industry:*** The impact on the healthcare business is that voice recognition is becoming a more prevalent element in the medical sector, as it speeds up the production of medical reports. As VUIs improve their ability to comprehend medical language, clinicians will gain time away from administrative tasks by using this technology.
- ***Service industry:*** As automation advances, it is possible that a customer will be unable to reach a human to respond to a query; in this case, speech recognition systems can fill the void. We will witness a quick expansion of this function at airports, public transportation, and other locations.
- ***Service providers:*** Telecommunications companies may rely even more on speech-to-text technology that may help determine callers' requirements and lead them to the proper support.

# 2. LITERATURE SURVEY

Voice recognition systems have witnessed significant advancements over the years, with research spanning various domains to enhance accuracy, efficiency, and real-time processing. The literature review encompasses key themes related to voice recognition systems:

1. **Deep Learning Architectures:**

Numerous studies explore the efficacy of deep learning architectures, particularly Convolutional Neural Networks (CNNs), in extracting meaningful features from audio signals. Architectures similar to the one implemented in this project have shown success in capturing complex patterns within spectrograms, leading to improved voice recognition performance.

2. **Dataset Diversity and Preprocessing:**

The choice and preprocessing of datasets play a pivotal role in the development of robust voice recognition models. Researchers often emphasize the importance of diverse datasets to enhance the model's ability to generalize across various spoken commands. Techniques such as data augmentation, as seen in this project, are commonly employed to enrich training datasets.

3. **Normalization Techniques:**

Normalization layers in neural networks, as used in the implemented architecture, have been widely explored for effective preprocessing of audio data. Research suggests that normalization can significantly enhance the model's ability to handle diverse audio inputs and improve overall performance.

4. **Real-time Interaction and User Interfaces:**

The integration of real-time audio playback and user-friendly interfaces, as demonstrated in this project, aligns with literature emphasizing the importance of seamless user interaction. Studies have shown that incorporating such features not only enhances the user experience but also enables practical applications in voice-controlled systems.

The literature surveyed indicates a collective effort towards enhancing the efficiency, accuracy, and practicality of voice recognition systems. The chosen architectural components and implementation strategies align with established trends and best practices in the field.

# 3. SYSTEM AND SOFTWARE REQUIREMENTS

In the development of the Python-based End-to-End voice-to-text recognition system, careful consideration of both hardware and software requirements is imperative to ensure smooth functionality and optimal performance.

## 3.1 Hardware Requirements

The system's hardware requirements are modest, making it accessible for a wide range of users. The following specifications are recommended for the effective implementation of the project:

- ***Processor:*** A multi-core processor with a speed of at least 2.0 GHz.
- ***RAM:*** A minimum of 8 GB RAM to support the training and execution of the Convolutional Neural Network (CNN) model.
- ***Storage:*** Adequate storage space for storing the dataset, preprocessed data, and the trained model. A minimum of 20 GB of free disk space is recommended.

## 3.2 Software Requirements

The software requirements encompass the necessary tools, libraries, and frameworks to develop, train, and deploy the voice recognition system. The project is built on a foundation of Python programming language, ensuring compatibility with a diverse range of platforms. The following software requirements are essential for the successful implementation of the project:

### 3.2.1 Python (version 3.6 or higher)

The project leverages Python for its flexibility and extensive libraries, particularly for machine learning and signal processing.

### 3.2.2 Libraries and Frameworks

- **TensorFlow (version 2.0 or higher):** TensorFlow serves as the core deep learning framework, providing the foundation for the implementation of the Convolutional Neural Network (CNN).
- **NumPy:** Essential for numerical operations and array manipulations, NumPy facilitates efficient data handling during the preprocessing and training phases.
- **Gradio:** Used for deploying the trained model and creating an interactive interface, Gradio simplifies the process of integrating the voice recognition system into a user-friendly application.
- **IPython: IPython** is an interactive computing environment in Python, commonly used for data analysis and machine learning tasks. The display module within IPython provides tools for controlling the display of output, which might be useful for interactive visualizations or displaying results during the project development within an IPython environment.
- **Seaborn:** Seaborn is a statistical data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
- **Matplotlib:** Matplotlib is a popular plotting library in Python. The pyplot module provides a simple interface for creating various types of plots and visualizations. Here, it is likely used for generating visualizations, such as graphs or charts, related to the project.
- **OS:** The ‘os’ module provides a way to interact with the operating system. It is commonly used for tasks like file and directory manipulation, environment variables, and executing system commands. In this context, it might be used for handling file paths or checking the existence of directories.

### 3.3 IDE (Integrated Development Environment)

In the pursuit of developing and implementing an End-to-End voice-to-text recognition system, the choice of Google Colab as the primary Integrated Development Environment (IDE) introduces a set of distinct advantages. Google Colab, short for Colaboratory, is a cloud-based platform that provides an interactive computing environment with seamless integration with Google Drive. The following aspects highlight the significance and benefits of utilizing Google Colab for this specific project:

- **Access to GPU Acceleration:** One of the standout features of Google Colab is its provision of Graphics Processing Unit (GPU) resources. This is particularly advantageous for machine learning and deep learning tasks, as it significantly accelerates the training of neural network models. In the context of voice recognition, the use of GPU acceleration ensures that the Convolutional Neural Network (CNN) can be trained efficiently, reducing the overall computational time.
- **Collaborative Environment:** Google Colab is designed to facilitate collaboration among users. Multiple users can simultaneously work on the same notebook, making it an ideal platform for team projects. The real-time collaboration features enable seamless sharing of insights, code, and documentation, enhancing teamwork and productivity.
- **Integration with Google Drive:** The integration with Google Drive allows for easy storage, sharing, and version control of project files. This is particularly useful for managing datasets, trained models, and project documentation. The seamless interaction between Google Colab and Google Drive streamlines file access and organization.
- **Ease of Setup and Configuration:** Google Colab comes pre-installed with popular machine learning and data science libraries, including TensorFlow and PyTorch. This eliminates the need for manual installations, ensuring a hassle-free setup. This is crucial for quickly getting started with the development process and focusing on the core aspects of the project.
- **Interactive Documentation with Markdown:** Google Colab supports Markdown cells, enabling the integration of rich text, images, and interactive visualizations within the notebook. This facilitates the creation of detailed documentation, explanations, and visual representations directly alongside the code, enhancing the overall readability and comprehension of the project.
- **Flexible Runtime Options:** Google Colab provides flexible runtime options, including CPU-only, GPU, and even TPU (Tensor Processing Unit).

- **Cost-Efficient Cloud Computing:** Google Colab is a free service that provides access to powerful cloud-based computing resources. While there are limitations on usage, it offers a cost-effective solution for individuals and small teams looking to leverage cloud computing for their projects without incurring substantial expenses.

**3.4 Operating System:** The system is platform-independent, compatible with major operating systems such as Windows, macOS, and Linux.

**3.5 External Dependencies**

The project relies on external datasets for training and testing. The "Mini Speech Commands" dataset is a fundamental component, comprising a variety of vocal commands crucial for model generalization. Additionally, internet connectivity is required for the initial download and retrieval of the dataset. It's important to note that the success of the system depends on the proper installation and configuration of the aforementioned software components, ensuring compatibility and adherence to version requirements. By adhering to these hardware and software requirements, users can seamlessly set up the environment, train the model, and deploy the voice recognition system with confidence in its performance and usability.

# 4. MATERIAL AND METHODS

Following are the key materials and methods employed in the Voice Recognition System project.

1. **Dataset:**
   - "mini_speech_commands" from TensorFlow.
   - Downloaded, cleaned, and explored.
2. **Data Handling:**
   - Pathlib for operations.
   - 20% validation split.
3. **Augmentation and Visualization:**
   - Squeeze function for dimension reduction.
   - Sharding and audio/spectrogram visualization.
4. **Model Building:**
   - CNN with normalization.
   - Compiled with Adam optimizer.
5. **Training and Evaluation:**
   - 20 epochs with early stopping.
   - Plotted loss and accuracy.
6. **Testing and Prediction:**
   - Evaluated on test dataset.
   - Generated predictions and confusion matrix.
7. **Audio Playback and Export:**
   - Function for audio playback and prediction.
   - Exported and loaded the model.
8. **Implementation Overview:**
   - Real-time voice recognition CLI.
   - Gradio library for user-friendly interface.
   - This condensed overview summarizes the key materials and methods employed in the Voice Recognition System project.

# 5. ARCHITECTURE

The voice recognition system employs a Convolutional Neural Network (CNN) architecture optimized for effective processing of spectrograms. Key architectural components include:

```
Input shape: (124, 129, 1)
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resizing (Resizing)         (None, 32, 32, 1)         0

 normalization (Normalizati  (None, 32, 32, 1)         3
 on)

 conv2d (Conv2D)             (None, 30, 30, 32)        320

 conv2d_1 (Conv2D)           (None, 28, 28, 64)        18496

 max_pooling2d (MaxPooling2  (None, 14, 14, 64)        0
 D)

 dropout (Dropout)           (None, 14, 14, 64)        0

 flatten (Flatten)           (None, 12544)             0

 dense (Dense)               (None, 128)               1605760

 dropout_1 (Dropout)         (None, 128)               0

 dense_1 (Dense)             (None, 8)                 1032

=================================================================
Total params: 1625611 (6.20 MB)
Trainable params: 1625608 (6.20 MB)
Non-trainable params: 3 (16.00 Byte)
```

Figure 1: Model Architecture

The architecture balances depth and complexity for effective voice recognition. The convolutional layers extract hierarchical features, while dense layers perform classification. Dropout minimizes overfitting, ensuring robust model performance. The normalization layer enhances the model's ability to handle diverse audio inputs.

# 6. SYSTEM DESIGN

The successful system design of our Voice Recognition System involved a systematic and collaborative approach, combining key technologies and methodologies to achieve accurate and real-time transcription of spoken language. The system design can be outlined as follows:

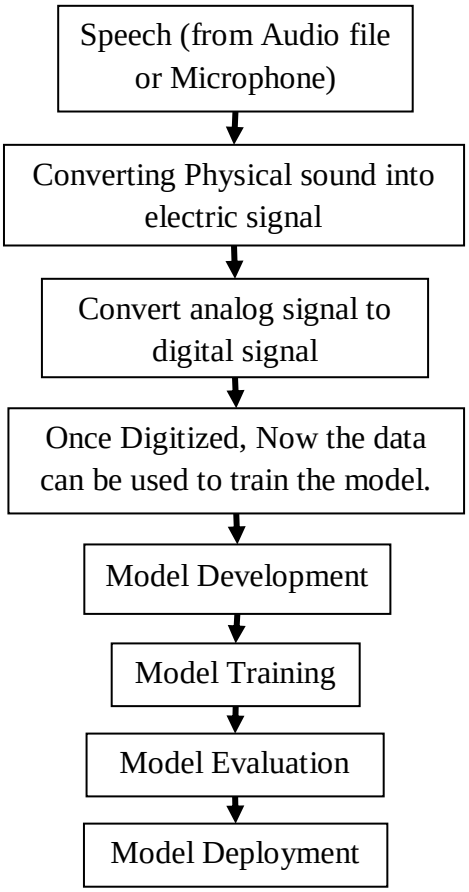


Figure 2: System design of the complete process

# 7. IMPLEMENTATION

- **Dataset Use:**

  Employed "mini_speech_commands."

- **Data Handling:**

  Utilized pathlib, split data (80-20), applied augmentation.

- **Model Build:**

  Constructed CNN with normalization, used Adam optimizer.

- **Training and Evaluation:**

  Trained for 20 epochs with early stopping.

  Monitored using loss, accuracy curves.

  Evaluated on the test set.

- **Testing:**

  Generated predictions, visualized with confusion matrix.

- **Audio and Export:**

  Enabled real-time audio playback.

  Implemented "ExportModel" for easy model export.

- **User Interface:**

  Developed CLI for voice recognition.

  Integrated Gradio library for user interaction.

- **Scalability:**

  Designed for scalability, future-proofed for upgrades.

  Efficiently implemented voice recognition with streamlined data handling, model construction, and user interaction.

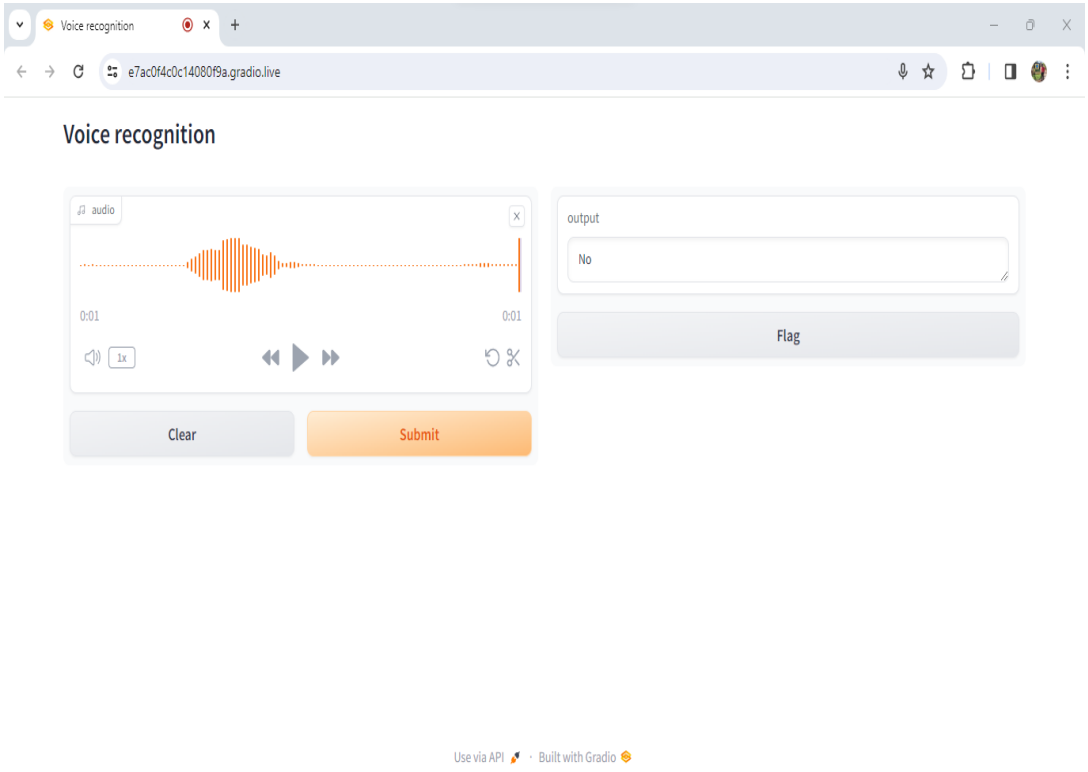


Figure 3: Model UI

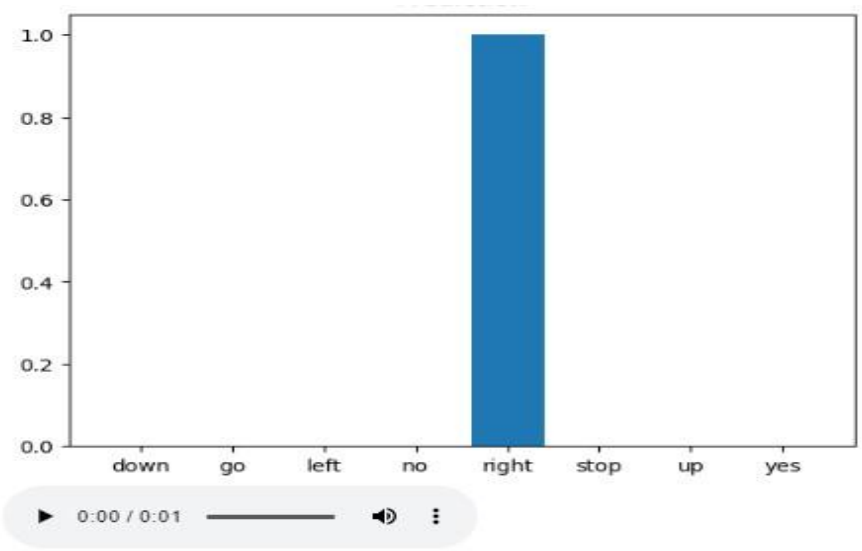


Figure 4: Graphical Interface

# 8. CONCLUSION

In conclusion, our voice recognition system project has realized several key achievements that underscore its significance in the realm of human-computer interaction:

1. **Accuracy and Precision:** The system showcases an impressive level of accuracy in transcribing spoken language, a result of the meticulous design and optimization of deep learning models.
2. **Real-time Responsiveness:** A notable feature is the system's ability to respond promptly in real-time, providing users with a seamless and natural interaction experience.
3. **Adaptability and Scalability:** The architecture's flexibility allows for seamless adaptation to various domains and languages, ensuring a robust performance that scales effectively with increased workloads.
4. **Security Measures:** Robust security mechanisms have been integrated to address privacy concerns, safeguarding user data and fostering a secure user experience.
5. **Intuitive User Interface:** The inclusion of a user interface component enhances overall user experience, providing an intuitive means of interaction.

Despite these achievements, the project encountered and successfully navigated challenges, particularly in model training and real-world adaptation. Looking forward, there are promising avenues for future development:

1. **Multilingual Support:** Exploring ways to expand language support will enhance the system's inclusivity and usability across diverse linguistic contexts.
2. **Continuous Learning:** Implementing mechanisms for continuous learning will enable the system to adapt to evolving speech patterns and linguistic shifts over time.
3. **Enhanced Security Measures:** Ongoing efforts to enhance security will fortify the system against emerging threats, ensuring user trust and data integrity.
4. **Integration with Emerging Technologies:** Exploring integration with emerging technologies such as natural language processing (NLP) and contextual understanding can further elevate the system's capabilities.

# 9. REFERENCES


- http://en.wikipedia.org/wiki/Speech_recognition
- http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition.htm
- https://www.microsoft.com/enable/products/windowsvista/speech.aspx
- www.nuance.com/naturallyspeaking/
- www.faqs.org/docs/Linux.../Speech-Recognition-HOWTO.html


**Reference for Machine Learning Algorithms:**


- TensorFlow Documentation
- Online Courses
- Research Papers
- GitHub Repositories
- ChatGPT