

A Real-Time Conversational AI Voice Agent with Intelligent Speech Processing

Sanika Rajesh Dhage , Dr.Arshiya A.Khan

MCA(Computer Science) , Assistant Professor, Department of Computer Science

JSPM University,Wagholi,Pune,India.

sanikadhage1105@gmail.com

Abstract

This paper presents the design and development of a real-time AI-based voice agent system that can handle live voice conversations accurately and with minimal delay. The system uses a browser-based interface for easy user interaction and FastAPI as the backend framework to manage communication and processing tasks. Real-time audio streaming and processing are managed through a dedicated agent built on LiveKit. The system integrates the Google Gemini Realtime API for speech recognition and intelligent response generation. It also supports continuous streaming of logs, conversation transcripts, and system updates using Server-Sent Events (SSE). The architecture is designed to be modular, scalable, and efficient, making it suitable for applications such as AI assistants, customer support services, and other voice-based interactive systems.

I. INTRODUCTION

A. Introduction

In recent years, Artificial Intelligence (AI) and Natural Language Processing (NLP) have greatly improved the way humans interact with computers. Voice-based systems have become a popular and convenient method of communication because they allow users to interact naturally through speech. Applications such as virtual assistants and real-time communication platforms have demonstrated how voice technology can improve user experience and accessibility. However, many traditional voice systems depend heavily on cloud-based processing, which can lead to delays in communication, higher latency, and limited flexibility for customization. These limitations reduce their effectiveness in

applications that require continuous, interactive, and real-time voice communication.

B. Background

Although voice technology has advanced significantly, several challenges still remain. Many existing voice recognition systems experience delays that affect real-time communication. They may also have difficulty handling background noise and accurately identifying speech in changing environments. Another challenge is that many current systems lack a modular and scalable design, making them difficult to modify or expand for different applications. In addition, many solutions do not support event-driven communication, which is important for real-time monitoring, logging, and system interaction. To address these issues, there is a need for a real-time AI-based voice agent system that offers low-latency processing, improved speech recognition accuracy, and a scalable architecture capable of supporting real-time communication and monitoring.

C.Objectives

- The primary objectives of this research work are as follows:
- To design and develop a real-time AI-based voice agent system
- To achieve low-latency processing for seamless voice interaction
- To integrate intelligent speech understanding and response generation using AI models

- To implement effective voice activity detection and noise reduction techniques
- To build a modular and scalable architecture for real-time applications

D. Scope of the Work

The scope of the proposed system is limited to the development of a web-based real-time voice interaction platform. The system focuses on processing live audio input through a browser interface and generating intelligent responses using AI-based models. The implementation is designed for desktop environments and requires an active internet connection for accessing real-time AI services. The system emphasizes real-time performance, modular architecture, and efficient communication, making it suitable for applications such as AI assistants, interactive voice systems, and real-time support platforms.

II. LITERATURE REVIEW

Recent research in speech processing has focused on making voice-based systems more accurate and reliable in real-world environments. Many studies have worked on improving speech recognition, speaker identification, and performance in noisy surroundings.

In [1], X. Yu and colleagues developed a multi-speaker speech processing system that works effectively in noisy environments with different dialects. The system uses techniques such as data augmentation, speaker intelligent response generation, and continuous communication. Technologies such as FastAPI, LiveKit, and real-time AI APIs are used to provide low-latency and efficient voice interactions.

III. SYSTEM DESIGN AND IMPLEMENTATION

The proposed system is a real-time AI-powered voice assistant that can understand and respond to user queries quickly and efficiently. It combines speech detection, speech recognition, language processing, and speech synthesis into a single system to provide smooth and natural communication. [1,2,3,4,5]

diarization, and Large Language Models (LLMs) to improve speech recognition performance. It also uses Retrieval-Augmented Generation (RAG) and context-aware processing to enhance accuracy. Experimental results showed that the system reduced the Diarization Error Rate (DER) by up to 15%, improving performance when multiple speakers are present. However, the study mainly focuses on speech recognition and does not provide strong support for real-time communication. [1]

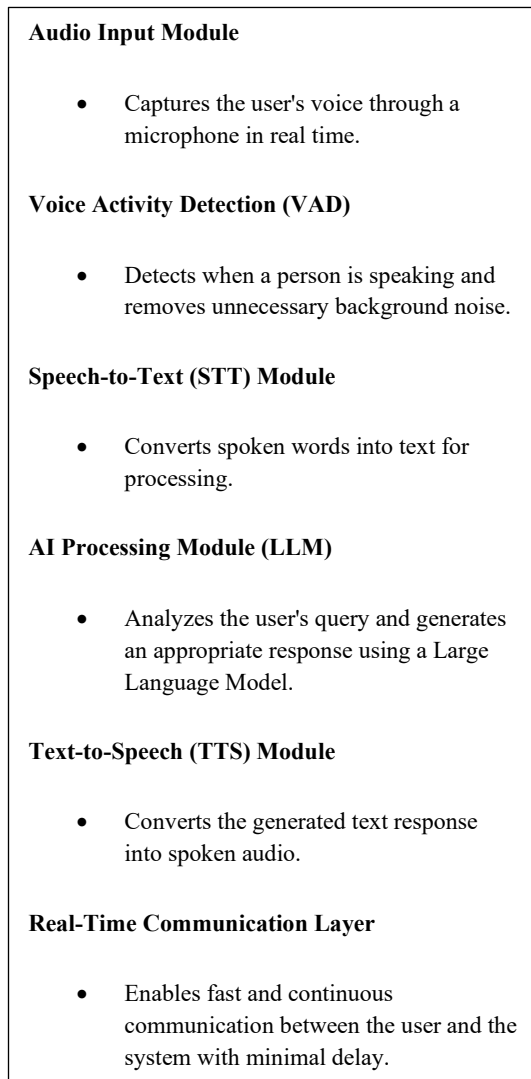
In [2], H. Jafarzadeh Asl and co-authors proposed a lightweight Voice Activity Detection (VAD) model for AI voice assistants and IoT devices. The model is designed to detect human speech accurately even in noisy environments. By using effective pre-processing and post-processing methods, the system improves voice detection without increasing computational complexity. Although the model performs well in speech detection, it does not cover the complete workflow of a real-time voice interaction system. [2]

Despite these developments, many existing systems still face issues such as processing delays, limited scalability, and lack of real-time communication support. Most studies focus on individual components like speech recognition or voice detection rather than building a complete integrated system. [4]

To overcome these limitations, the proposed system introduces a scalable and modular AI-based voice assistant that combines real-time audio processing,

A. System Architecture

The system consists of the following major components:



Audio Input Module

The Audio Input Module captures the user's voice through a microphone in real time. It acts as the first step

of the system by receiving spoken commands or queries from the user and sending them for further processing.

Voice Activity Detection (VAD)

The Voice Activity Detection (VAD) module identifies when a user is speaking and separates speech from background noise. This helps the system process only relevant audio, improving accuracy and efficiency.

Speech-to-Text (STT) Module

The Speech-to-Text (STT) module converts the user's spoken words into text format. This text can then be understood and processed by the AI system.

AI Processing Module (LLM)

The AI Processing Module uses a Large Language Model (LLM) to analyze the user's query. It understands the meaning of the text and generates an appropriate and intelligent response.

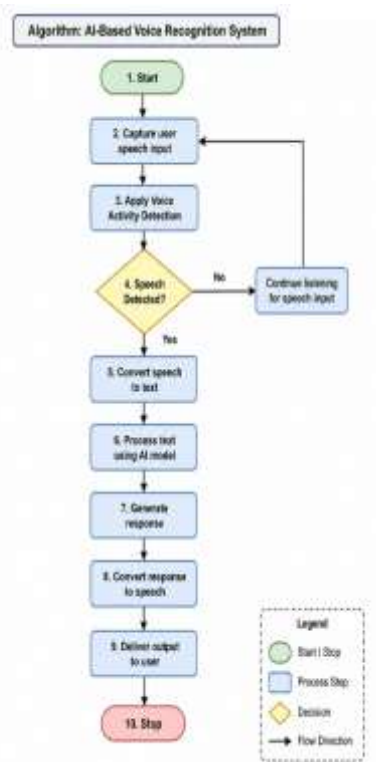
Text-to-Speech (TTS) Module

The Text-to-Speech (TTS) module converts the AI-generated text response into natural-sounding speech. This allows the system to communicate with users through voice.

Real-Time Communication Layer

The Real-Time Communication Layer ensures smooth and fast interaction between the user and the system. It minimizes delays and enables continuous communication for a better user experience. [1,2,4]

B.ALGORITHM



Algorithm Explanation:

Step 1: Start

The system starts and initializes all required modules for voice processing.

Step 2: Capture User Speech Input

The user's voice is captured through a microphone in real time.

Step 3: Apply Voice Activity Detection (VAD)

The system analyzes the audio input to identify speech and filter out background noise.

Step 4: Speech Detected?

A decision is made to check whether valid speech is present.

- No: The system continues listening for voice input.
- Yes: The process moves to the next step.

Step 5: Convert Speech to Text

The detected speech is converted into text using a Speech-to-Text (STT) module.

Step 6: Process Text Using AI Model

The text is analyzed by the AI model to understand the user's request and determine an appropriate response.

Step 7: Generate Response

Based on the processed input, the AI generates a suitable text response.

Step 8: Convert Response to Speech

The generated text response is converted into spoken audio using a Text-to-Speech (TTS) module.

Step 9: Deliver Output to User

The audio response is delivered to the user through speakers or headphones.

Step 10: Stop

The process ends after the response is provided to the user.[2,4]

C. System Flow

The system operates through a sequence of steps that enable real-time interaction between the user and the voice assistant. When the user speaks into the microphone, the Voice Activity Detection (VAD) module identifies the presence of speech and filters out silence or unnecessary background noise. The detected speech is then converted into text using the Speech-to-Text (STT) module and forwarded to the AI model for processing.

The AI model analyzes the user's query and generates an appropriate response based on the input. This response is then converted into speech using the Text-to-Speech (TTS) module and delivered to the user through the output device. By continuously processing input and output in real time, the system provides fast, accurate, and smooth communication.[1,3,5]

D. System Tools

System Architecture of Real-Time Voice Assistant using FastAPI and LiveKit



1. User Voice Input

The process begins when the user speaks through the browser using a microphone.

The voice input is captured and sent for further processing.

2. LiveKit Real-Time Communication

LiveKit establishes a real-time connection between the user and the backend system.

It ensures smooth and continuous audio transmission with low delay.

3. FastAPI Backend

FastAPI acts as the main backend server of the system. It manages requests and coordinates communication between all components.

3. Limited Speech Recognition Accuracy

Recognition accuracy may decrease when users have different accents, speaking styles, or pronunciations.

4. Silero Voice Activity Detection (VAD)

The Silero VAD module detects when the user is speaking.

It removes silence and reduces unnecessary background noise before processing.

5. Gemini API Processing

The cleaned speech data is sent to the Gemini API.

The API understands the user's query and generates an appropriate response.

6. Response Handling by FastAPI

FastAPI receives the generated response from the Gemini API.

The response is prepared for delivery in text or audio form.

7.Voice Output to the User

The final response is sent back to the user's browser.

IV.PROBLEM STATEMENT

Despite the rapid growth of voice recognition technologies, several challenges continue to affect the performance and usability of existing voice-based systems.

1. High Response Time (Latency)

Many traditional voice processing systems require significant time to process user input and generate responses. This delay reduces the quality of interaction and makes such systems less suitable for real-time applications where immediate feedback is expected.

2. Difficulty Handling Background Noise

Existing systems often struggle to recognize speech accurately in noisy environments. Background sounds such as traffic, conversations, and television audio can interfere with speech recognition, leading to incorrect interpretations and responses.

These variations can make it difficult for the system to correctly understand spoken commands.

4. Lack of Scalable Architecture

Many voice-based applications are designed for specific use cases and cannot efficiently support increasing workloads or a growing number of users. This limits their ability to expand and adapt to future requirements.

5. Poor Modularity and Flexibility

Traditional systems often consist of tightly coupled components, making modifications and upgrades difficult. Adding new features or improving existing functionality may require extensive changes throughout the system.

6. Limited Real-Time Monitoring and Communication

Many existing solutions lack efficient real-time communication mechanisms between system components. This makes it difficult to monitor processing activities, track system status, and provide instant updates to users.

7. Need for an Efficient Real-Time Voice System

To address these challenges, there is a need for an intelligent voice recognition system that offers low latency, effective noise reduction, accurate speech recognition, and real-time communication. The system should also be scalable, modular, and flexible enough to support future enhancements. In addition, it should be capable of handling different user accents and speaking styles to improve recognition accuracy. The system must provide quick and reliable responses to ensure a smooth user experience. It should efficiently process voice input even in noisy environments and maintain consistent performance. Furthermore, the design should allow easy integration of new technologies and features as user requirements evolve over time.

Summary

The primary objective is to develop a real-time voice recognition system that delivers fast response times, improved speech recognition accuracy, efficient noise handling, and a flexible architecture. Such a system can enhance user experience and support a wide range of real-world voice-based applications.

V. RESULT AND DISCUSSION

The proposed AI-based Voice Recognition System was tested in a real-time environment to evaluate its overall performance. Different test cases were conducted using multiple users and various background noise conditions to check the system's accuracy, speed, and reliability.

The test results showed that the system is capable of processing voice input and generating responses with very low delay. This allows users to interact with the system smoothly and naturally. The Silero Voice Activity Detection (VAD) module successfully identified speech segments and filtered out silence or unnecessary sounds, improving the quality of speech processing.

The Google Gemini AI model was able to understand user queries effectively and generate relevant and meaningful responses. The responses were delivered quickly, making the interaction more efficient and user-friendly.

The integration of LiveKit BVC for noise cancellation significantly improved audio quality by reducing background noise and unwanted sounds. This helped the system recognize speech more accurately, even in environments with moderate noise.

A comparison with traditional voice recognition systems showed that the proposed architecture provides better real-time performance and flexibility. The modular design makes the system easier to maintain and expand with additional features in the future. Furthermore, the event-driven communication mechanism enables faster updates and smoother interaction between system components.

The system also performed well when multiple users interacted with it, maintaining stable performance and quick response times. Overall, the results demonstrate that the proposed system is reliable, scalable, and suitable for real-world voice-based applications where accurate speech recognition and real-time communication are required.[1,2,4]

VI. CONCLUSION

The proposed real-time AI voice assistant successfully combines speech recognition, intelligent response generation, and noise reduction into a single system. It overcomes many limitations of traditional voice-based systems by providing fast, reliable, and real-time communication.

By using technologies such as FastAPI, LiveKit, Silero VAD, and the Google Gemini API, the system is able to process voice input accurately and generate responses with minimal delay. This makes it suitable for various applications, including virtual assistants, customer support services, and interactive voice-based platforms.

The system performs effectively even in noisy environments, improving the overall user experience. Its scalable and flexible design also provides a strong foundation for future improvements and the addition of more advanced features.

VII. ACKNOWLEDGEMENT

We would like to express our heartfelt thanks to our project guide and mentors for their continuous guidance, support, and encouragement throughout the development of this project. Their valuable suggestions and assistance helped us complete the work successfully.

We also appreciate the resources, documentation, and tools provided by FastAPI, LiveKit, and the Google AI team. These resources played an important role in understanding the technologies and implementing the system effectively.[1,2,3,4,5]

VIII. REFERENCES

- [1] X. Yu, Y. Ju, X. Yuan, L. Yang and H. Zhang, "A Multi-Speaker Speech Processing System for Dialect- Rich Environments with LLM-Based Optimization," 2025 9th IEEE International Conference on Network Intelligence and Digital Content (IC-NIDC), Beijing, China, 2025, pp. 213-217, doi: 10.1109/IC-NIDC67200.2025.11390222.
- [2] H. Jafarzadeh Asl, M. Ghazvini Nejad, A. Edraki, M. Asgharian and V. Partovi Nia, "Tiny Noise-Robust Voice Activity Detector for Voice Assistants," 2025 IEEE 35th International Workshop on Machine Learning for Signal Processing (MLSP), Istanbul, Turkiye, 2025, pp. 1-6, doi: 10.1109/MLSP62443.2025.11204268. keywords: {Voice activity detection;Accuracy;Personal voice assistants;Transforms;Noise robustness;Background noise;Human voice;Noise measurement;Internet of Things;Smart glasses;Noise Robustness;Speech Recognition;Voice activity Detection},I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
- [3] M. Yamano, Z. Song and J. Hoshino, "Augmenting the Social Presence of Interactive Characters Using Real- time Speech Recognition," 2022 Nicograph International (NicoInt), Tokyo, Japan, 2022, pp. 85-88, doi: 10.1109/NicoInt55861.2022.00024.
- [4] K. N., R. V., S. S. S. and D. R., "Intelligent Personal Assistant - Implementing Voice Commands enabling Speech Recognition," 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2020, pp. 1-5, doi: 10.1109/ICSCAN49426.2020.9262279.
- [5] R. Manimegalai, M. Shobana, C. Rajesh, C. Parthasarathy, A. V. Sriharsha and T. D. Kumar, "Adaptive Emotion-Aware Speech Synthesis on Edge Devices: A Lightweight Dynamic Model Adaptation Framework using LoRA and OnDevice Fine-Tuning," 2026 International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2026, pp. 1513-1520, doi: 10.1109/ICEARS67481.2026.11416528. keywords: {Training;Adaptation models;Renewable energy sources;Quantization (signal);Speech enhancement;Real-time systems;Physiology;Speech synthesis;Reliability;Robots;Speech Synthesis;Edge Devices;Low-Rank Adaptation;Lightweight Neural Adapters;On-Device Learning}