AI Voice Chat Agent on Raspberry Pi

Andreh Samaan

09.05.2025

1 Introduction

This report details the development of an AI voice chat agent implemented on a Raspberry Pi. The primary goal of this project was to explore the diverse functionalities of the Raspberry Pi by creating an interactive voice-based chatbot. Users can speak to the agent via a microphone, and the agent, powered by a local AI model, processes the input and responds audibly. While an initial objective included integrating an AI camera for visual input, time constraints prevented its implementation. This project demonstrates the feasibility of running sophisticated AI and natural language processing tasks on a resource-constrained platform like the Raspberry Pi.

2 Results

The project successfully achieved several key milestones, culminating in a functional voice chat agent with a graphical user interface.

2.1 Milestone 1: Speech to Text

The initial stage involved setting up speech recognition capabilities on the Raspberry Pi. The 'SpeechRecognition' library was utilized to capture audio input from a microphone and transcribe it into text using Google's Speech Recognition service. While this requires an internet connection, the speed and accuracy were deemed acceptable for the project's current state.

2.2 Milestone 2: Text to Speech

The second milestone focused on enabling the agent to provide audible responses. The 'gTTS' library was employed to convert text output from the AI model into speech. While acknowledging that more natural-sounding text-to-speech options exist, 'gTTS' was chosen for its relatively fast processing time, which is crucial for maintaining a somewhat real-time conversational flow.

2.3 Milestone 3: Local AI Chat (Terminal)

Before integrating voice capabilities, the core logic of interacting with a local AI model was established. The ‘ollama‘ library was used to interface with the ‘gemma3:1b‘ language model. This milestone involved sending text prompts to the AI via the terminal and receiving text-based responses, ensuring the basic AI interaction was functional locally.

2.4 Milestone 4: Integrated Voice Chat

This crucial step involved connecting the speech-to-text and text-to-speech modules with the local AI. User voice input was transcribed to text, sent as a prompt to the Ollama model, and the AI's text response was then converted to speech and played back to the user. This integration formed the core functionality of the voice chat agent.

2.5 Milestone 5: Graphical User Interface

To enhance user interaction, a graphical user interface (GUI) was developed using the ‘tkinter‘ library. The GUI features a chat history display, showing both user input and AI responses, and a ”Record” button to initiate voice input. The use of threading ensures that the GUI remains responsive while the AI processes information and generates responses.

3 Highlight: Functional Voice Chat with GUI

The most significant achievement of this project is the successful creation of a functional voice chat agent with a user-friendly graphical interface. While the core AI model (Gemma 3, 1B) runs locally, both the speech-to-text (via Google's service) and text-to-speech (via gTTS) functionalities require an active internet connection. This project demonstrates the potential of leveraging open-source libraries and local AI models, in conjunction with online services when needed, for interactive applications.

4 Conclusions

The AI Voice Chat Agent project successfully met its primary goal of exploring the Raspberry Pi's capabilities through the development of an interactive voice chatbot. The integration of speech-to-text, a local AI model, and text-to-speech into a cohesive application with a GUI represents a significant accomplishment.

While the current implementation has limitations, such as the reliance on an internet connection for speech recognition and the somewhat less natural-sounding text-to-speech, it provides a solid foundation for future improvements. Addressing the latency and naturalness of the AI's spoken responses, perhaps by exploring more sophisticated text segmentation techniques or faster, higher-quality TTS engines, would be a valuable next step. The initial ambition of incorporating an AI camera for visual interaction remains a potential avenue for future development, which could significantly enhance the agent's capabilities and user experience.

Overall, this project serves as a valuable learning experience, showcasing the potential of the Raspberry Pi for AI and interactive applications and highlighting areas for further exploration and refinement.

Disclaimer Regarding AI Assistance

During the course of this project, AI tools were utilized to enhance the development process. These tools provided assistance with writing and understanding code snippets, exploring different implementation strategies, and refining the language and structure of this report to ensure clarity and professionalism.