RoboPi A Fully Offline Voice-Interactive Desktop Companion Robot

N.V. Sravan Kumar[1], Parupally Chandana[2], Nikhil S Kallakuri[3], Ch. Raja[4]
[1,2,3]*Undergraduate Students, Department of Electronics and Communication Engineering, Mahatma Gandhi Institute of Technology, Telangana, India.*
[4]*Associate Professor, Department of Electronics and Communication Engineering, Mahatma Gandhi Institute of Technology, Telangana, India*
doi.org/10.64643/IJIRTV12I12-201309-459

***Abstract*—This paper presents RoboPi, a fully offline voice-interactive AI companion robot developed using Raspberry Pi 5. The system is designed to provide natural human-like interaction without relying on cloud services, ensuring complete data privacy. RoboPi integrates advanced technologies including local large language models (LLMs) using Ollama, speech recognition through Whisper, and voice synthesis using Piper TTS. The system also features a visual feedback mechanism through a TFT display that shows real-time facial expressions based on interaction states. The robot is capable of understanding voice commands, generating contextual responses, and expressing emotions through a multi-modal interface. The implementation demonstrates that advanced AI functionalities can be efficiently executed on resource-constrained hardware. RoboPi can be used as a personal assistant, educational tool, or smart home interface. The project highlights the potential of embedded AI systems in creating intelligent and privacy-focused applications.**

***Index Terms*—Offline AI, Voice Assistant, Raspberry Pi, Embedded Systems, Local LLM, Human-Computer Interaction.**

I. INTRODUCTION

In recent years, artificial intelligence-based assistants such as Alexa and Google Assistant have become widely popular. However, these systems depend heavily on cloud infrastructure, raising concerns about privacy, latency, and internet dependency.

To address these challenges, this project introduces RoboPi, an offline AI companion robot capable of performing voice interactions without internet connectivity. The system leverages local processing to ensure user data remains secure while providing real-time responses [1].

The main objective is to create an intelligent, user-friendly robotic assistant that can communicate naturally and provide visual feedback through facial expressions. This project demonstrates the integration of embedded systems with AI technologies to create a compact and efficient solution.

II. LITERATURE REVIEW

"In recent years, artificial intelligence-based voice assistants have gained significant popularity [2-4]". This project addresses these limitations by integrating multiple lightweight AI technologies into a unified offline system, enabling real-time interaction without internet connectivity [5]. In recent years, artificial intelligence-based voice assistants have gained significant popularity [6-8]. Systems such as Amazon Alexa, Google Assistant, and Apple Siri provide efficient human-computer interaction using cloud-based processing [9-11]. However, these systems depend on continuous internet connectivity and raise concerns related to data privacy and latency.

To overcome these limitations, researchers have focused on developing offline AI systems using edge computing [12-13]. Edge-based systems process data locally, reducing response time and ensuring better data security. However, implementing such systems on embedded platforms remains challenging due to limited computational resources [14-16].

Speech recognition technologies like Whisper have improved the accuracy of converting speech to text, even in noisy environments [17-18]. Similarly, text-to-speech systems such as Piper provide natural voice

output suitable for embedded applications. Recent advancements in local large language models (LLMs) have enabled devices to generate human-like responses without relying on cloud services [19-20].
Despite these developments, most existing systems focus on individual components rather than providing a fully integrated solution [21-23]. The proposed RoboPi system addresses this gap by combining speech recognition, AI processing, voice synthesis, and visual feedback into a single offline platform, ensuring efficient and secure interaction.

III. SYSTEM ARCHITECTURE

The architecture of the RoboPi system is designed to ensure efficient coordination between hardware and software components. Each module is developed independently and integrated through a centralized control mechanism.
The voice input module continuously monitors the environment for user interaction. It uses a USB microphone to capture audio signals and forward them to the processing unit. The system is designed to handle real-time audio input with minimal delay.
The processing unit plays a crucial role in the system. It performs speech-to-text conversion and processes the user query using a local large language model. The use of optimized models ensures that the system maintains a balance between performance and resource utilization.
The voice output module is responsible for converting the generated text response into speech. It uses a text-to-speech engine to produce clear and natural audio output. The audio is then played through a speaker connected via an amplifier.
The visual feedback system enhances interaction by displaying facial expressions on a TFT screen. These expressions change dynamically based on system states such as listening, thinking, and speaking. This feature improves user engagement and provides intuitive feedback.
The system also includes a state management mechanism that synchronizes all modules. This ensures that voice output and visual expressions occur simultaneously without delay.
Overall, the architecture is flexible and scalable, allowing future enhancements such as additional sensors, camera modules, or IoT integration.

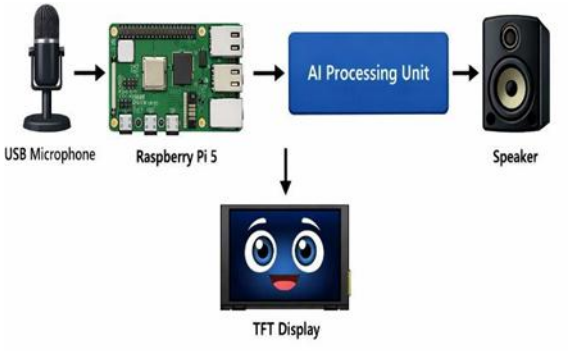


Figure.1: System Architecture of RoboPi robot.

Above figure 1 shows the overall system architecture of the RoboPi robot. It illustrates the flow of voice input from the microphone to the Raspberry Pi for processing. The AI model generates a response which is converted into speech and output through the speaker. The TFT display provides visual feedback through facial expressions, enhancing user interaction.

IV. METHODOLOGY

The methodology of the RoboPi system focuses on integrating multiple AI and embedded system components to achieve seamless human-robot interaction. The overall workflow is divided into four major stages: voice input processing, AI-based response generation, voice output synthesis, and visual feedback rendering.

A. Voice Input Processing

The system continuously listens for user interaction through a USB microphone. A wake word detection mechanism is implemented to activate the system only when required, reducing unnecessary processing and power consumption. Once the wake word is detected, the system records the user's voice input for a short duration.
The recorded audio is then processed using a speech recognition model to convert it into text format. This step is optimized to handle variations in speech, including different accents and background noise, ensuring reliable performance in real-world environments.

B. AI-Based Response Generation

The converted text is passed to the AI processing module, where a local large language model generates a response. The system uses optimized models to balance response speed and accuracy. A smaller model

is used for quick responses, while a larger model is used for more complex queries.

A custom conversational framework is implemented to maintain context across multiple interactions. This allows the system to generate more meaningful and relevant responses. Additionally, a personality-based prompt system is used to give the robot a consistent and engaging communication style.

C. Voice Output Synthesis

The generated text response is converted into speech using a text-to-speech engine. The system produces natural and clear audio output, enhancing the overall user experience. The audio generation process is optimized to minimize delay and ensure real-time interaction.

The synthesized speech is played through a speaker connected via an amplifier. Proper synchronization is maintained to ensure that audio output aligns with system states and user expectations.

D. Visual Feedback Rendering

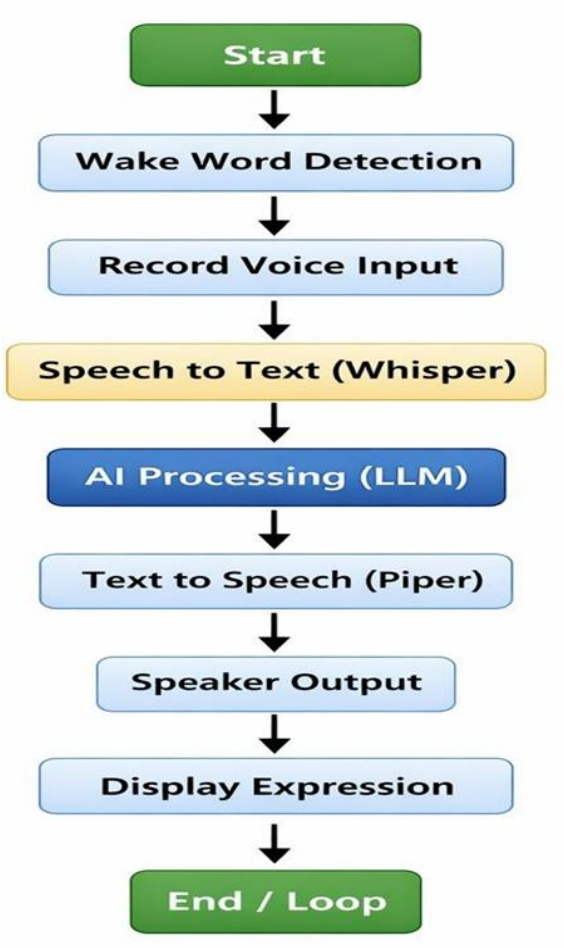
To improve interaction, the system includes a visual feedback mechanism that displays facial expressions on a TFT screen. These expressions represent different states such as idle, listening, thinking, speaking, and happy.

The facial expressions are generated using lightweight image processing techniques, ensuring minimal computational overhead. The system updates the display in real time, synchronized with voice input and output processes.

E. System Integration and Control

All modules are integrated using a central control system running on Raspberry Pi 5. The system follows a continuous loop operation, where it listens for input, processes data, and generates output without interruption.

Error handling mechanisms are implemented to ensure system stability. In case of failures, the system can recover without affecting overall functionality. This robust design ensures reliable and efficient operation.



The above flowchart shows the working flow of the RoboPi system. It starts with wake word detection followed by recording the user's voice input. The input is converted into text, processed using an AI model, and then converted back into speech. Finally, the output is delivered through the speaker along with visual feedback on the display, completing the interaction loop.

V. IMPLEMENTATION

The implementation of the RoboPi system involves the integration of both hardware and software components to achieve a fully functional offline AI companion robot. The development process follows a structured approach, starting from hardware setup to software development and final system integration.

A. Hardware Implementation

The hardware setup is centered around the Raspberry Pi 5, which acts as the main processing unit. It is equipped with 4GB RAM, making it capable of

handling AI workloads efficiently. A USB microphone is used for capturing voice input, while a 3W speaker connected through a MAX98357A I2S amplifier is used for audio output.

A 2.4-inch TFT LCD display is integrated to provide visual feedback in the form of facial expressions. The display communicates with the Raspberry Pi through an SPI interface, ensuring fast data transfer. A reliable power supply of 5V 5A is used to maintain stable system performance. Additionally, an active cooling system with a heat sink and fan is employed to prevent overheating during continuous operation.

All hardware components are carefully connected and tested individually to ensure proper functionality before system integration.

Figure.2: Hardware setup of the RoboPi system.

Above figure 2 shows the hardware setup of the RoboPi system. It includes the Raspberry Pi 5 as the central processing unit connected with essential components. The USB microphone is used for capturing voice input, while the speaker provides audio output. The TFT display is integrated to show facial expressions, enabling visual interaction.

B. Software Implementation

The software is developed using Python 3.11 due to its simplicity and wide range of library support. The system uses multiple libraries and frameworks to implement different functionalities.

The Ollama framework is used for running local large language models, enabling AI-based conversation without internet dependency. Whisper is used for speech-to-text conversion, while Piper TTS is used for generating natural speech output. OpenWakeWord is integrated for wake word detection, allowing hands-free operation.

For visual feedback, the Pillow library is used to generate facial expressions, and display control is handled using appropriate drivers. GPIO libraries are used for interfacing with hardware components.

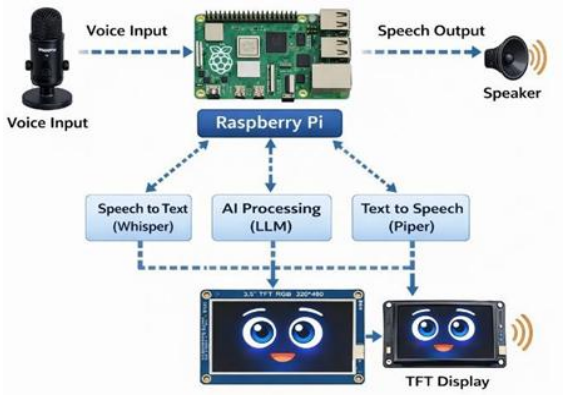


Figure.3: Software setup of the RoboPi system.

Above figure 3 shows the software setup of the RoboPi system. It illustrates the interaction between different software modules such as speech-to-text, AI processing, and text-to-speech. The Raspberry Pi acts as the central controller managing all operations. The processed output is delivered through the speaker and displayed on the TFT screen for user interaction.

C. Modular Development Approach

The system is developed using a modular approach, where each functionality is implemented as a separate module. This includes voice input, AI processing, voice output, and display modules.

Each module is tested independently to ensure correctness before integration. This approach simplifies debugging and allows easy modification or upgrades in the future.

D. System Integration

After individual modules are developed and tested, they are integrated into a single system. The integration is performed in stages to ensure stability. The system operates in a continuous loop, where it listens for user input, processes the request, and generates output.

Synchronization between modules is carefully handled to ensure smooth operation. For example, facial expressions are updated in real time based on the system state, and audio output is coordinated with AI

responses.

E. Testing and Validation

The system is tested under different conditions to ensure reliability. Continuous interaction, different voice inputs, and system load conditions are evaluated. Performance metrics such as response time, memory usage, and temperature are monitored. The system successfully demonstrates stable operation with efficient resource utilization.

VI. RESULTS AND DISCUSSIONS

The RoboPi system was tested extensively to evaluate its performance, efficiency, and reliability under different operating conditions. The evaluation focused on key parameters such as response time, memory usage, CPU temperature, speech quality, and overall user interaction experience.

A. Performance Evaluation

The system demonstrated efficient real-time performance during testing. The response time varied depending on the AI model used. The lightweight model (qwen2.5:0.5b) provided faster responses within 1–2 seconds, making it suitable for quick interactions and demonstrations. On the other hand, the larger model (gemma2:2b) produced more accurate and context-aware responses, with a response time of approximately 3–4 seconds.

Memory utilization was observed to range between 600MB and 2GB during AI processing. Despite the limited hardware resources, the system maintained stable performance without crashes or significant slowdowns. This indicates effective resource management and optimization.

B. Thermal and System Stability Analysis

Temperature monitoring was conducted to ensure safe operation of the Raspberry Pi 5. The system maintained an idle temperature of around 42°C, which increased to 60–70°C under continuous processing load. With the use of an active cooling system, the temperature remained within safe operating limits.

The system operated continuously without unexpected shutdowns, demonstrating good stability. Proper power supply and cooling mechanisms contributed to consistent performance.

C. Speech Recognition and Output Quality

The speech recognition system performed accurately in various environments, including moderate background noise. The Whisper model successfully converted spoken input into text with high accuracy. The text-to-speech output generated using Piper TTS was clear and natural. The pronunciation and tone were consistent, providing a pleasant user experience. The average processing time for generating speech output was less than one second, ensuring minimal delay.

D. Visual Feedback Performance

The visual feedback system effectively displayed different facial expressions corresponding to system states such as listening, thinking, speaking, and idle. The expressions were updated in real time with minimal delay, enhancing user interaction.

The use of lightweight image rendering ensured that the display module did not significantly impact overall system performance. The visual interface contributed to a more engaging and intuitive communication experience.

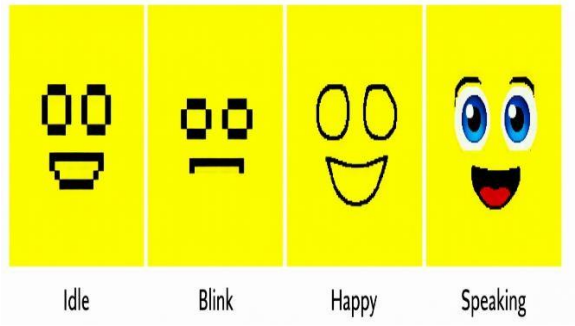


Figure 4: Facial expression outputs of the RoboPi system.

Above figure 4 shows the facial expression outputs of the RoboPi system. It includes different states such as idle, blinking, happy, and speaking expressions. These expressions are displayed on the TFT screen based on the system's current activity. This visual feedback enhances user interaction and makes the communication more intuitive.

E. Comparative Analysis

A comparison between the two AI models highlighted the trade-off between speed and accuracy. The smaller model offered faster responses with lower resource usage, while the larger model provided better

contextual understanding at the cost of increased processing time.

This flexibility allows the system to adapt based on user requirements, making it suitable for both real-time interaction and more complex conversational tasks.

F. Overall System Performance

Overall, the RoboPi system successfully achieved its objective of providing a fully offline, voice-interactive AI experience. The integration of multiple components resulted in a smooth and efficient workflow.

The system proved to be reliable, responsive, and user-friendly. The combination of voice interaction and visual feedback created a more natural and engaging human-robot interaction environment.

VII. ADVANTAGES AND LIMITATIONS

A. Advantages

The RoboPi system offers several significant advantages due to its offline architecture and efficient integration of AI technologies.

One of the primary advantages is complete offline operation, which eliminates the need for continuous internet connectivity. This makes the system highly reliable in environments with limited or no network access.

Another important benefit is enhanced data privacy and security. Since all processing is performed locally on the device, user data is not transmitted to external servers, reducing the risk of data breaches and unauthorized access.

The system also provides low latency interaction. Unlike cloud-based systems that depend on network communication, RoboPi processes inputs locally, resulting in faster response times and improved user experience.

The integration of multi-modal interaction (voice and visual feedback) improves communication between the user and the system. The facial expressions displayed on the TFT screen make the interaction more engaging and intuitive.

Additionally, the system is compact and portable, making it suitable for personal use, educational purposes, and research applications. The modular design allows easy customization and future upgrades.

The use of open-source tools and frameworks reduces development cost and makes the system accessible for students and developers.

B. Limitations

Despite its advantages, the RoboPi system has certain limitations that need to be considered.

The primary limitation is the restricted computational capability of the embedded hardware. Running large AI models on a Raspberry Pi with limited memory can affect performance and restrict the complexity of tasks.

Another limitation is the trade-off between speed and accuracy. Smaller models provide faster responses but may lack depth in understanding, while larger models offer better accuracy at the cost of increased processing time.

The system also faces memory and storage constraints, which limit the number and size of AI models that can be deployed locally.

Thermal management is another concern, as continuous processing can lead to increased temperature. Although cooling solutions are used, prolonged usage may still impact performance.

The current system has limited contextual awareness compared to advanced cloud-based AI systems. It may not handle highly complex queries as effectively as larger models running on powerful servers.

Additionally, the system depends on external hardware components such as microphones, speakers, and displays, which may affect overall reliability if not properly configured.

C. Discussion

The advantages and limitations highlight the trade-offs involved in designing an offline AI system. While the system ensures privacy and independence from cloud services, it requires careful optimization to achieve acceptable performance.

Overall, RoboPi successfully demonstrates that a balance between functionality, performance, and resource constraints can be achieved using efficient design and implementation strategies.

VIII. FUTURE SCOPE

The RoboPi system provides a strong foundation for further enhancements in the field of embedded AI and human-robot interaction. Although the current system

demonstrates efficient offline functionality, several improvements can be implemented to extend its capabilities.

One of the major future enhancements is the integration of computer vision techniques. By adding a camera module, the robot can perform tasks such as face recognition, object detection, and gesture recognition. This would significantly improve interaction by enabling visual understanding of the environment.

Another important area of improvement is the development of emotion-aware AI systems. By analyzing voice tone or facial expressions, the robot can detect user emotions and respond accordingly, making interactions more natural and personalized.

The system can also be extended to function as a smart home controller by integrating IoT devices. This would allow users to control appliances such as lights, fans, and security systems using voice commands, making RoboPi a central hub for home automation.

In addition, the integration of a mobile or web application can enhance user accessibility. Users can monitor and control the robot remotely, update system settings, and manage data more efficiently.

Further research can focus on optimizing AI models to achieve better performance within limited hardware resources. Techniques such as model compression, quantization, and efficient memory management can improve speed and reduce power consumption.

Another potential improvement is the introduction of a hybrid system, where the robot operates offline by default but can optionally connect to cloud services when needed for advanced processing. This would combine the benefits of both offline and online systems.

Finally, the system can be expanded for real-world applications such as elderly assistance, education, healthcare support, and customer service robots. With continuous advancements, RoboPi has the potential to evolve into a highly intelligent and versatile robotic assistant.

IX. CONCLUSION

This paper presented the design and development of RoboPi, a fully offline voice-interactive AI companion robot built using Raspberry Pi 5. The system successfully integrates multiple technologies including speech recognition, natural language processing, text-to-speech synthesis, and visual feedback into a single embedded platform.

The proposed system addresses key limitations of existing cloud-based voice assistants by eliminating dependency on internet connectivity and ensuring complete data privacy. By processing all interactions locally, RoboPi provides a secure and reliable solution for real-time human-robot interaction.

The implementation demonstrates that advanced AI functionalities can be efficiently executed on resource-constrained hardware through proper optimization and model selection. The use of lightweight models enables a balance between performance and accuracy, making the system suitable for practical applications.

Experimental results confirm that the system achieves stable performance with acceptable response time, efficient memory usage, and effective thermal management. The integration of visual feedback further enhances user engagement by providing intuitive and expressive communication.

Overall, RoboPi represents a significant step towards the development of intelligent, privacy-focused, and standalone AI systems. The project highlights the potential of embedded AI in transforming everyday human-computer interaction.

“RoboPi represents a significant step toward the development”. The proposed system can serve as a foundation for future research and development in areas such as personal robotics, smart home automation, and assistive technologies. With further improvements and enhancements, RoboPi has the potential to evolve into a highly capable and versatile AI companion.

REFERENCES

[1] H. Touvron et al., “LLaMA 2: Open Foundation and Fine-Tuned Chat Models,” arXiv preprint arXiv:2307.09288, 2023.

[2] A. Radford et al., “Robust Speech Recognition via Large-Scale Weak Supervision,” arXiv preprint arXiv:2212.04356, 2022.

[3] Google Team, “Gemma: Open Models Based on Gemini Research and Technology,” arXiv preprint arXiv:2403.08295, 2024.

[4] Raspberry Pi Foundation, “Raspberry Pi 5 Documentation,” 2024.

[5] Ollama, “Running Large Language Models

Locally," 2024.
[6] Rhasspy Project, "Piper Text-to-Speech System," 2024.
[7] Broadcom Inc., "BCM2712 Technical Reference Manual," 2024.
[8] Maxim Integrated, "MAX98357A I2S Amplifier Datasheet," 2023.
[9] ILI Technology Corp., "ILI9341 TFT LCD Controller Datasheet," 2023.
[10] Brenpoly, "Be-More-Agent: BMO AI Agent on Raspberry Pi," GitHub, 2024.
[11] Dwain Barnes, "On-Device Raspberry Pi Voice Assistant," GitHub, 2024.
[12] OpenAI,"Whisper: Robust Speech Recognition," GitHub, 2024.
[13] Python Software Foundation, "Python 3.11 Documentation,"2024.
[14] Adafruit Industries, "CircuitPython RGB Display Library," 2024.
[15] Pillow Library, "Python Imaging Library Documentation," 2024.
[16] Anthropic,"Claude AI Assistant Documentation," 2024.
[17] Debian Project, "Debian GNU/Linux Documentation," 2024.
[18] Linux Foundation, "I2S Audio Interface Specification," 2024.
[19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, no. 7553, pp. 436–444,2015.
[20] K. Hwang and M. Chen, "Big-Data Analytics for Cloud, IoT, and Cognitive Computing," Wiley, 2017.
[21] W. Shi et al., "Edge Computing: Vision and Challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, 2016.
[22] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8,pp. 1735–1780, 1997.
[23] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.