Offline Hindi Voice Assistant on Raspberry Pi 4

Penta Leela Prakash, Kanuboyina Jyothirganesh, Kornala Charan Tej
Guided by: Dr. B.V.V. Satyanarayana

February 11, 2026

1 Project Introduction

The goal of this project is to develop a privacy-focused, offline Hindi voice assistant. By utilizing the Raspberry Pi 4, we eliminate the need for cloud-based processing, ensuring that user data remains local and secure. This is particularly relevant for regional languages where cloud support may be inconsistent or privacy-invasive.This also focus on latency and we aimed to get the latency below 2 seconds by prioritising the different tasks.

2 System Architecture

The system follows a sequential pipeline to process audio and generate responses entirely on the edge device.

The pipeline consists of:

- **Hardware:** Raspberry Pi 4 (4GB), USB Microphone, 3.5mm Speaker.
- **ASR Engine:** Wav2Vec2 (Fine-tuned for Hindi).
- **TTS Engine:** eSpeak-NG (Hindi Phoneme Support).

3 Methodology

3.1 Speech Recognition & Intent Recognition

Audio is sampled at 16kHz and passed through the Wav2Vec2 model. The resulting text is then parsed using a custom Python logic layer. This layer identifies intents such as time queries or weather updates through pattern matching.By considering the different synonyms and slangs we implemented a intent recognization process by making array of keyqords.

3.2 Performance Optimizations

To achieve a response time of approximately 3 seconds on a CPU-limited device, we implemented the following:

1. **Quantization:** Model weights were converted to INT8 to speed up mathematical operations.
2. **Multi-threading:** By using of pipelinig concept and threads Audio capture and inference run on separate threads to prevent input lag.

3. **Priority Scaling:** The inference process was assigned a high *Nice* value to prioritize CPU cycles and by making this we always search for input from the mic USB port.

4 Results and Performance Metrics

The following table highlights the operational efficiency of the system during testing with 15 standard Hindi commands:

Metric	**Value/Status**
Average Latency	3 Seconds
Word Error Rate (WER)	$\approx 15\%$
CPU Peak Usage	85%
RAM Usage	1.2 GB
Offline Capability	100% Functional

5 Challenges

- **Memory Footprint:** Managing the high memory demands of the transformer architecture.
- **Dialectal Variation:** Difficulty in recognizing various Hindi slangs and regional dialects.
- **Sentence Complexity:** Managing intent for complex sentence structures.
- **Acoustic Sensitivity:** Performance issues related to varying voice pitches and the need for high-quality audio input.

6 Conclusion

- **Hardware Viability:** Proves that with proper optimization, complex AI models can run on low-power hardware like the Raspberry Pi 4.
- **Optimization Techniques:** Integration of Quantization and Multi-threading was essential for performance.
- **Resource Management:** Maximum RAM utilization and lightweight ASR models facilitate efficient processing.
- **Offline Interaction:** Demonstrated the capability of localized human-AI interference without cloud dependency.