Offline Hindi Voice Assistant

An Embedded Edge AI System on Raspberry Pi

Kamalesh E
Ravi Prasath N K
Poorani R
Electronics and Communication Engineering
Government College of Technology,Coimbatore

Contents

1 Abstract

This report presents the design and deployment of a fully offline Hindi Voice Assistant implemented on Raspberry Pi. The system performs real-time speech recognition, intent classification, and speech synthesis without requiring internet connectivity.

The assistant is optimized for embedded edge deployment with low latency, minimal CPU utilization, controlled memory usage, and power-aware architecture. The system supports practical Hindi commands including time inquiry, date inquiry, contact lookup, nutrition queries, demo-based light control, and safe system shutdown.

2 Problem Statement

Most existing voice assistants depend heavily on cloud infrastructure, resulting in:

- Internet dependency
- Privacy concerns
- High power consumption
- Unsuitability for edge-only deployment

The objective of this project is to develop a fully offline Hindi voice assistant capable of operating entirely on Raspberry Pi hardware.

3 Design Objectives

- Develop real-time offline Hindi ASR.
- Achieve response latency below 1 second.
- Maintain CPU utilization below 5%.
- Limit RAM usage to approximately 300MB.
- Implement intelligent sleep mode for power optimization.
- Design modular and scalable architecture.
- Enable safe system shutdown via voice command.

4 System Architecture

4.1 Pipeline Overview

The assistant follows a modular processing pipeline:

1. Audio Capture Layer (16kHz Mono)
2. Vosk Speech Recognition Engine
3. Rule-Based Intent Recognition

4. State Management Controller
5. Response Generator
6. Text-to-Speech Engine (eSpeak-NG)
7. Bluetooth / USB Audio Output

4.2 Block Diagram Representation

```
Microphone
    ↓
Audio Stream (16kHz)
    ↓
Vosk ASR Engine
    ↓
Intent Recognition
    ↓
State Manager
    ↓
Response Generator
    ↓
eSpeak-NG TTS
    ↓
Audio Output
```

5 Core Technologies

Component	Technology
Programming Language	Python 3.11
ASR Engine	Vosk Offline
Audio Streaming	PyAudio
TTS Engine	eSpeak-NG
Service Management	systemd
Hardware Platform	Raspberry Pi

6 Implemented Commands

6.1 Information Retrieval

- Time Inquiry
- Date Inquiry
- Days in Month
- Weather (Offline Demo)

- Temperature (Demo)
- Nutrition Query

6.2 System Interaction

- Light ON / OFF (Demo)
- Contact Lookup
- Repeat Last Response
- Assistant Identity Query
- Safe System Shutdown

6.3 State Management

- Sleep Mode
- Wake Mode
- Automatic Sleep after Inactivity

7 State Machine Design

The assistant operates in two primary states:

AWAKE State

- Continuously listens for commands
- Processes input through ASR
- Generates speech output

SLEEP State

- Reduced listening activity
- Only wake command active
- Lower CPU consumption

State transitions occur after 5 minutes of inactivity or when a wake command is detected.

8 Software Optimizations

- 16kHz mono audio reduces processing load.
- Rule-based intent detection instead of heavy NLP models.
- RapidFuzz similarity matching for error tolerance.
- Microphone disabled during TTS to prevent feedback.
- Dictionary-based O(1) lookup for contacts and nutrition data.
- Clean state-machine architecture.
- Systemd auto-start configuration.

9 Hardware Optimizations

- Optimized for Raspberry Pi edge deployment.
- Bluetooth stabilization delay during boot.
- Low-power idle state (3W).
- Controlled memory footprint (250–300MB).
- Thread separation for ASR and TTS.

10 Performance Evaluation

Metric	Observed Performance
Average Latency	¡ 1 second
CPU Utilization	¡ 5%
RAM Usage	250–300 MB
Idle Power Consumption	3W

The system demonstrates stable real-time interaction under embedded hardware constraints.

11 Advantages

- Fully offline operation
- Privacy-preserving architecture
- Low latency response
- Power-efficient design
- Modular and extensible implementation

12 Limitations

- Limited vocabulary compared to cloud systems
- Accent sensitivity
- Demo-based hardware actions
- No deep contextual NLP understanding

13 Future Work

- Wake-word detection module
- ONNX-based ASR acceleration
- Real IoT sensor integration
- Multi-language support
- Hardware LED status indicators
- Quantized ASR model optimization

14 Conclusion

This project successfully demonstrates that a fully offline Hindi Voice Assistant can be deployed on Raspberry Pi with real-time performance, low computational overhead, and energy-efficient architecture.

The implementation validates the feasibility of privacy-preserving edge AI assistants for resource-constrained embedded systems.

Keywords: Edge AI, Offline ASR, Raspberry Pi, Embedded Systems, Hindi Voice Assistant, Low-Latency Systems