Bharat AI-SoC Student Challenge

Offline, Privacy-Preserving Hindi Voice Assistant on Raspberry Pi

1.Introduction

The Bharat AI-SoC Student Challenge promotes industry-ready skills through hands-on projects in Artificial Intelligence and SoC design using **Arm architecture**. It encourages students to build innovative and sustainable AI solutions for real-world applications.

As part of this initiative, this project develops a fully offline Hindi voice assistant on an Arm-based platform, integrating on-device ASR, intent processing, and TTS. The work highlights efficient edge AI deployment, privacy preservation, low-latency performance, and regional language accessibility on resource-constrained SoC systems.

2.Problem Statement

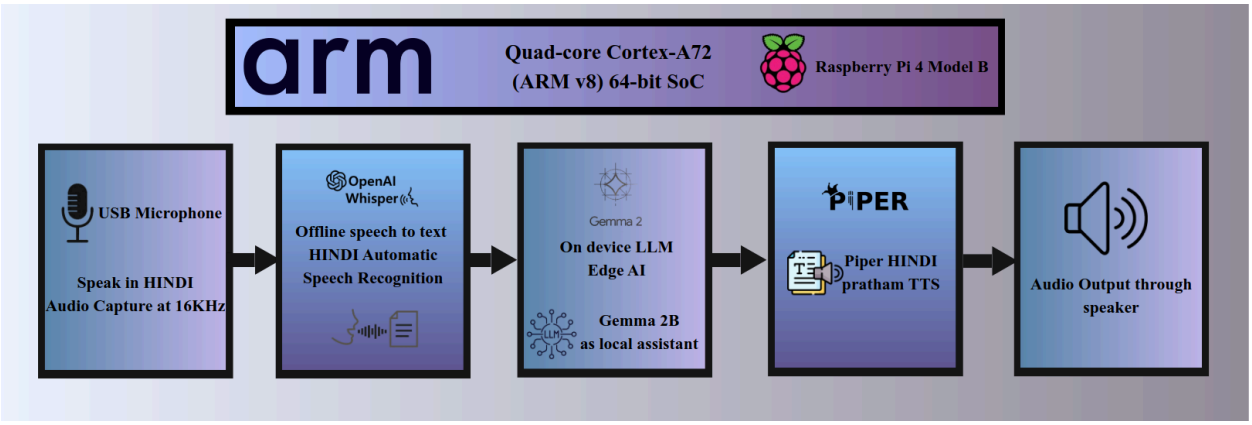
Most modern voice assistants depend on cloud-based processing, resulting in privacy concerns, network dependency, and higher response latency. These limitations restrict their use in secure, low-connectivity, and edge-computing environments-especially for regional languages like Hindi.

This project addresses the challenge of developing a fully offline, low-latency Hindi voice assistant on an Arm-based SBC such as the Raspberry Pi 4 or Raspberry Pi 5 using only CPU resources.

3.Objectives

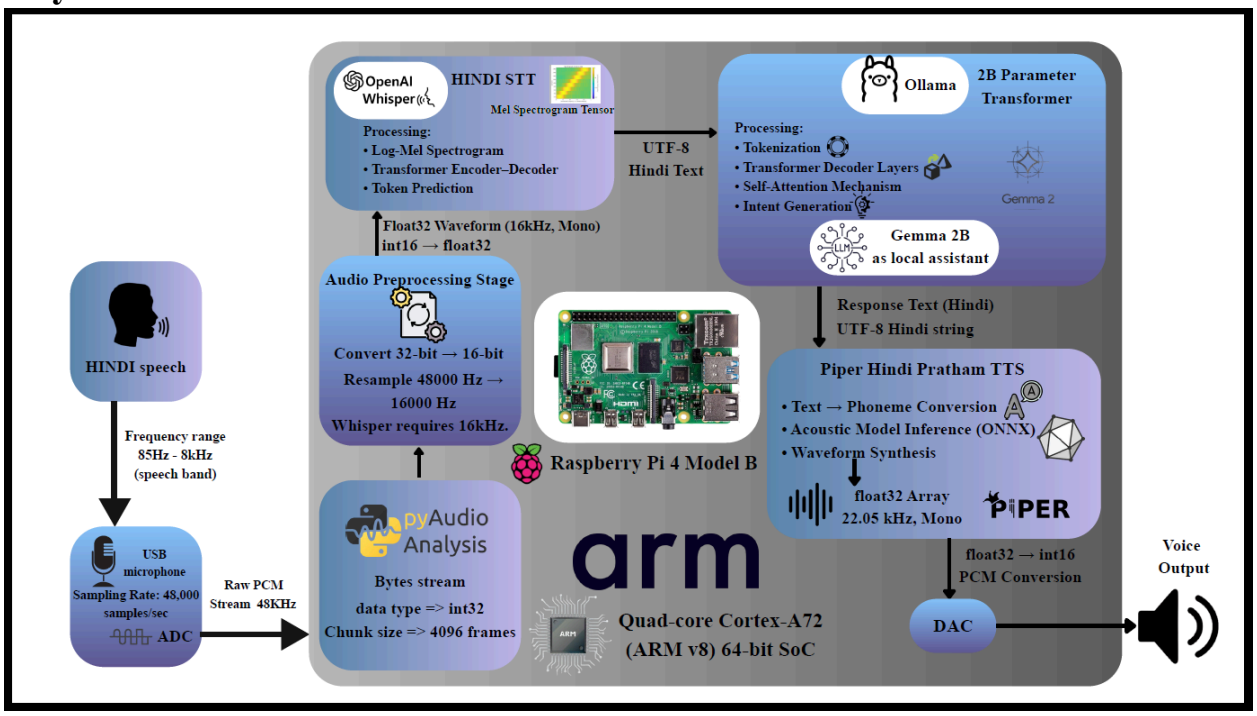
- Develop a fully offline Hindi voice assistant on an Arm-based Single Board Computer.
- Integrate on-device ASR, intent recognition, and TTS into a complete embedded pipeline.
- Enable efficient CPU-only processing without external accelerators.
- Ensure privacy-preserving operation and optimized resource utilization on constrained edge hardware.

4.Methodology



The system was implemented on the Raspberry Pi 4 Model B using a modular speech-processing pipeline. Audio input was captured through a USB microphone using Python and preprocessed for recognition. The Whisper model performed offline speech-to-text conversion, and the transcribed Hindi text was processed by the Gemma 2B model for intent recognition.A custom Python-based command layer mapped the detected intents to predefined system actions. The generated response was converted into speech using the Piper Hindi Pratham TTS engine and played back through the speaker. The complete pipeline was optimized to ensure efficient CPU utilization and stable offline operation.

5.System Architecture



- Speech is captured through the USB microphone.
- Whisper performs offline speech-to-text conversion.
- Gemma 2B processes the text for intent recognition.
- The command layer executes system-level actions.
- Piper Hindi Pratham generates the final speech output.
- The entire pipeline runs on CPU without cloud dependency.

6.Hardware Utilization

Hardware Used

- Board: Raspberry Pi 4 Model B
- Processor: Quad-core ARM Cortex-A72
- RAM: 8 GB LPDDR4
- Storage: microSD (32GB)
- Audio Input: USB Microphone (Shure MV7+)
- Audio Output: Bluetooth / 3.5mm speaker

Resource	Worst-Case Usage	Condition
CPU Usage	85–100% (all cores active)	During Whisper transcription or Gemma response generation
RAM Usage	4.5 – 6.5 GB	When ASR + LLM models are loaded simultaneously
Storage Usage	8 – 15 GB	Model weights + Python environment + dependencies
CPU Temperature	70–80°C	Continuous inference without active cooling
Audio Latency	Slight buffer delay	During simultaneous playback + inference

7.Optimization Techniques

- Selected Whisper Base model (~140MB) instead of larger models to reduce memory and CPU load.

- Used Piper Hindi medium voice (~60MB) to ensure lightweight TTS execution.

- Limited CPU threads using:
  OMP_NUM_THREADS=2
  OPENBLAS_NUM_THREADS=2
  to prevent OpenBLAS thread contention and system hanging.

- Recorded audio at native 48kHz (USB mic format) and performed software resampling to 16kHz, improving recognition accuracy.

- Converted audio from 32-bit to 16-bit PCM before model inference to ensure compatibility with Whisper.

- Increased PyAudio buffer size (CHUNK = 4096) to prevent input overflow and ALSA errors.

8.Applications & Future Scope

The offline Hindi voice assistant can be used in smart home systems, assistive technologies, educational tools, and public service kiosks, especially in areas with limited internet connectivity. Since the system operates entirely on-device, it exnsures privacy and low power consumption. In the future, the solution can be enhanced by supporting multiple Indian languages, integrating lightweight conversational models, reducing response latency through streaming recognition, and optimizing performance for large-scale edge deployments.

9.Conclusion

In this project, an offline, privacy-preserving Hindi voice assistant was successfully developed and deployed on a Raspberry Pi 4. The system integrates on-device speech recognition, intent processing, and text-to-speech synthesis without relying on cloud services. Through careful model selection and hardware optimization, the solution operates efficiently within the computational constraints of an ARM-based single-board computer. The project demonstrates the feasibility of implementing embedded speech AI for regional language applications while maintaining low power consumption, data privacy, and edge-device compatibility.

10.Team details

Team name : Cortex Crew

Team member 1 : Praveen R

Team member 2 : Ragul T

Team member 3 : S S JHOTHEESHWAR