Bharat AI-SoC Student Challenge 2026

Offline Privacy-Preserving Hindi Voice Assistant on Raspberry Pi

Problem Statement 1

R Hari Narayanan

Sreenand TM

Ramavarshini N

Amrita Vishwa Vidyapeetham, Chennai Campus

Abstract

This project presents an offline Hindi voice assistant implemented on a Raspberry Pi using Edge Artificial Intelligence techniques. Unlike conventional cloud-based voice assistants, the proposed system performs speech recognition, intent classification, and command execution locally, eliminating the need for continuous internet connectivity while ensuring low latency and enhanced user privacy.

The system integrates Vosk for offline speech recognition and FastText for intent classification using a custom-labeled Hindi command dataset. It supports multiple functionalities, including device control, mathematical operations, date and time queries, and personalized user interaction.

Model optimization techniques such as quantization are applied to reduce memory consumption and improve execution speed for embedded deployment. The implementation demonstrates efficient hardware–software co-design, real-time performance, and privacy-preserving AI execution on resource-constrained platforms. This work highlights the feasibility of deploying intelligent voice assistants on edge devices and contributes toward next-generation AI-enabled embedded systems.

1 Introduction

Voice assistants enable natural interaction between humans and machines. Most commercial voice assistants such as Amazon Alexa, Google Assistant, and Apple Siri rely on cloud-based processing, which introduces privacy risks, internet dependency, and increased response latency.

This project proposes an offline privacy-preserving Hindi voice assistant that performs speech recognition and decision-making locally using Edge Artificial Intelligence. Local processing eliminates data transmission to external servers and ensures faster response time.

The Bharat AI-SoC Student Challenge focuses on AI-driven System-on-Chip solutions integrating software and hardware design. This project addresses the challenge by implementing an embedded voice assistant capable of understanding Hindi commands and executing actions without internet connectivity.

Objectives of the Project

- Develop an offline Hindi speech recognition system
- Implement machine learning-based intent classification
- Enable device control and information retrieval
- Ensure privacy-preserving on-device processing
- Optimize models for embedded deployment

The system integrates speech recognition, natural language processing, and embedded hardware interaction to demonstrate efficient Edge AI deployment.

2 Background and Related Work

Speech recognition converts spoken language into textual representation using signal processing and machine learning techniques. Traditional systems rely on cloud infrastructure, while recent lightweight models enable embedded deployment.

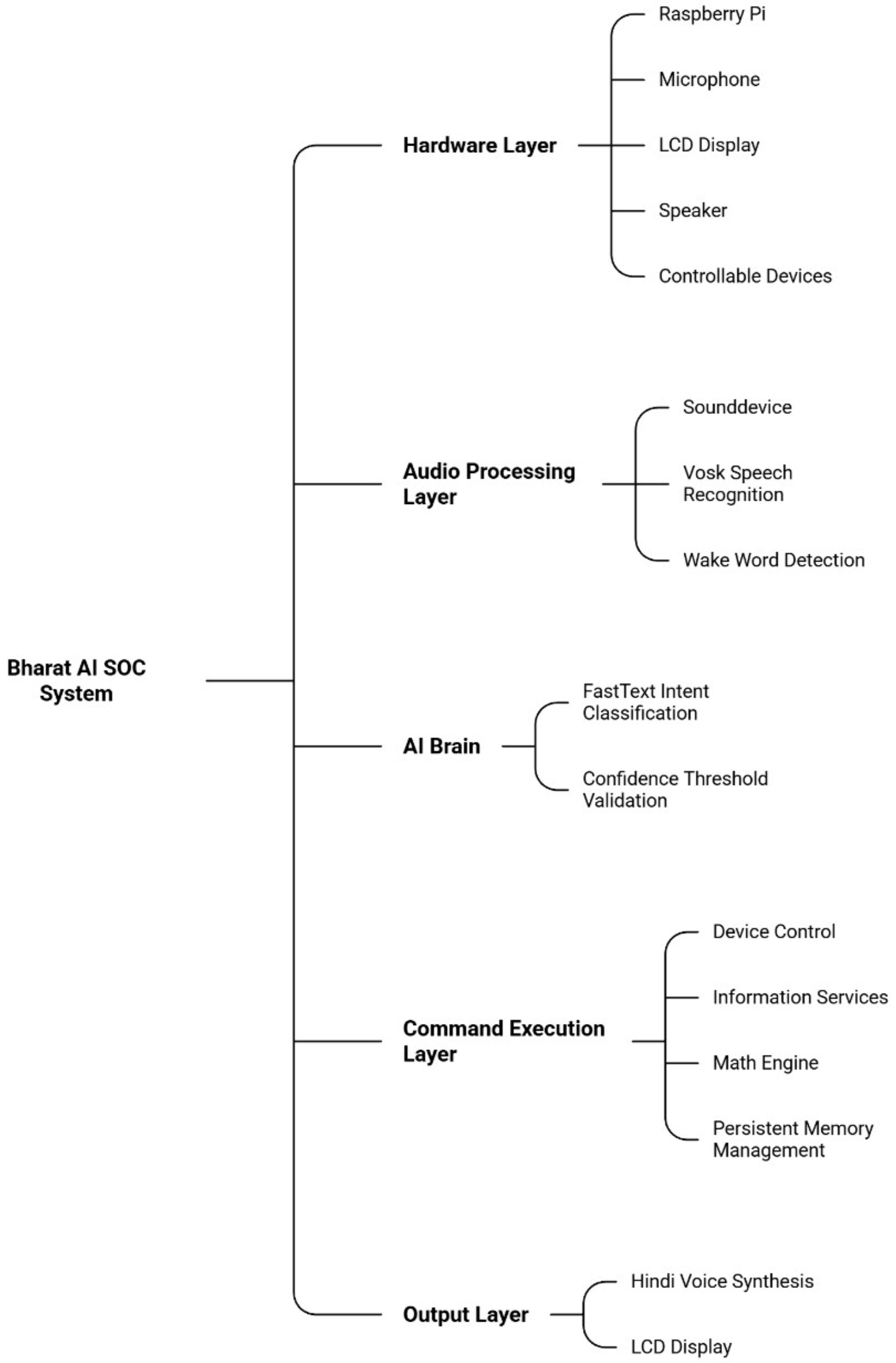
- **Vosk Toolkit:** Vosk is an open-source offline speech recognition engine supporting multiple languages with low computational requirements, making it suitable for Raspberry Pi.
- **FastText:** FastText is an efficient text classification library that uses shallow neural networks and n-gram features to provide fast inference with low memory usage.
- **Edge AI:** Edge AI performs computation locally on devices, improving privacy, reducing latency, and minimizing network usage.

This project integrates these technologies to build a fully offline embedded voice assistant.

3 System Architecture

The proposed system performs all operations locally on Raspberry Pi. The processing pipeline consists of the following stages:

- **Audio Input:** Voice commands are captured using a microphone.
- **Speech Recognition:** Vosk converts audio into text.
- **Intent Classification:** FastText predicts user intent from recognized text.
- **Command Execution:** Corresponding actions are performed.
- **Output Display:** Results are shown on an LCD display.


Bharat AI SOC System
Hardware Layer
Raspberry Pi
Microphone
LCD Display
Speaker
Controllable Devices
Audio Processing Layer
Sounddevice
Vosk Speech Recognition
Wake Word Detection
AI Brain
FastText Intent Classification
Confidence Threshold Validation
Command Execution Layer
Device Control
Information Services
Math Engine
Persistent Memory Management
Output Layer
Hindi Voice Synthesis
LCD Display

Supported Functionalities

- Fan and light control
- Time and date queries
- Mathematical operations
- User name storage and retrieval
- System control commands

The architecture ensures low latency and privacy-preserving processing.

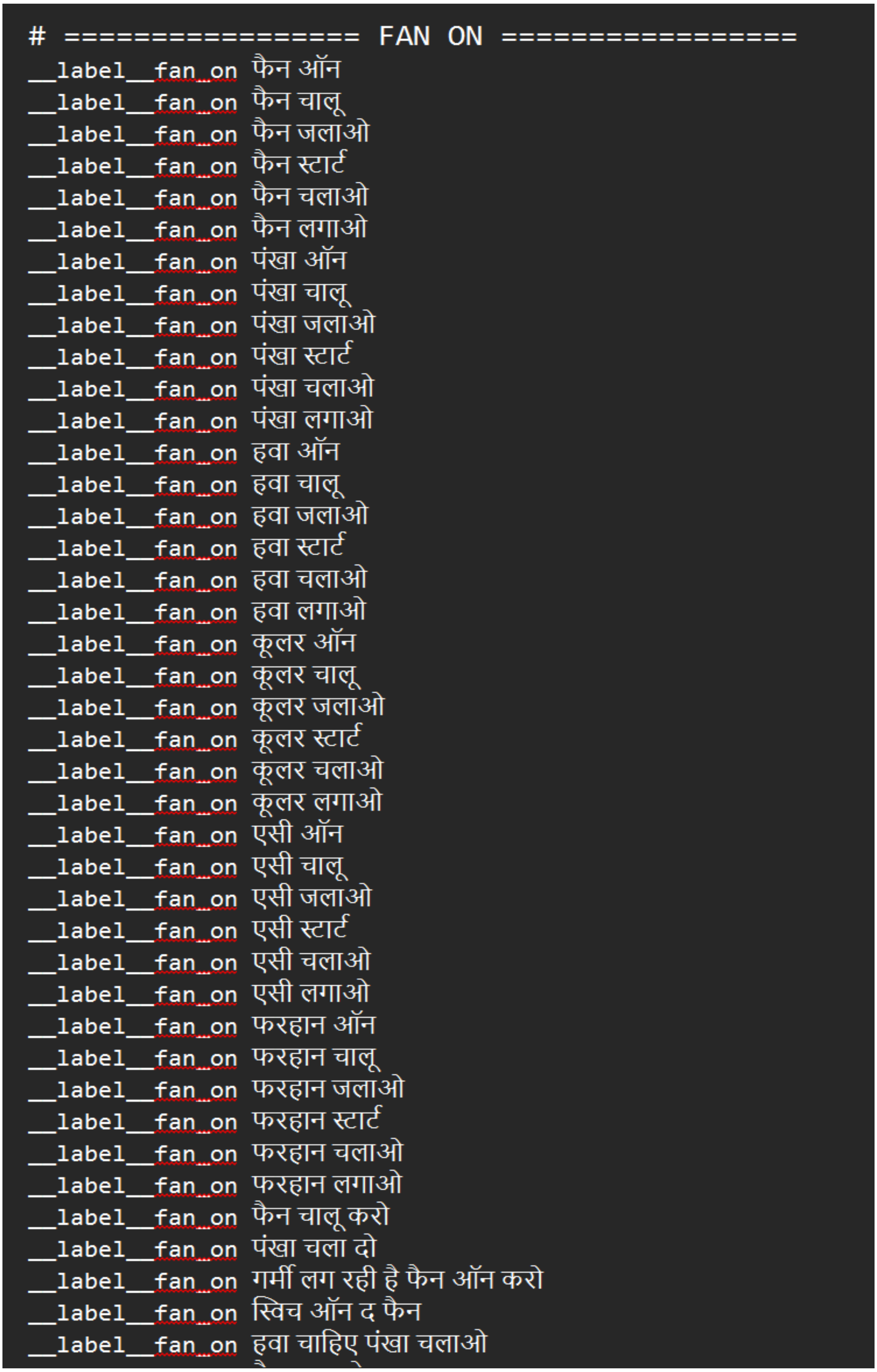
4 Dataset and Model Training

A custom labeled dataset containing Hindi command phrases was created for intent classification. Commands were mapped to predefined labels such as device control, time queries, and mathematical operations.

The FastText supervised learning model was trained using the following parameters:

- Learning rate: 1.0
- Training epochs: 25
- Word n-grams: 2
- Embedding dimension: 50

Multiple command variations were included to improve recognition accuracy. Fuzzy matching techniques were implemented to handle pronunciation variations and improve system robustness.



```
# ================== FAN ON ==================
__label__fan_on फैन ऑन
__label__fan_on फैन चालू
__label__fan_on फैन जलाओ
__label__fan_on फैन स्टार्ट
__label__fan_on फैन चलाओ
__label__fan_on फैन लगाओ
__label__fan_on पंखा ऑन
__label__fan_on पंखा चालू
__label__fan_on पंखा जलाओ
__label__fan_on पंखा स्टार्ट
__label__fan_on पंखा चलाओ
__label__fan_on पंखा लगाओ
__label__fan_on हवा ऑन
__label__fan_on हवा चालू
__label__fan_on हवा जलाओ
__label__fan_on हवा स्टार्ट
__label__fan_on हवा चलाओ
__label__fan_on हवा लगाओ
__label__fan_on कूलर ऑन
__label__fan_on कूलर चालू
__label__fan_on कूलर जलाओ
__label__fan_on कूलर स्टार्ट
__label__fan_on कूलर चलाओ
__label__fan_on कूलर लगाओ
__label__fan_on एसी ऑन
__label__fan_on एसी चालू
__label__fan_on एसी जलाओ
__label__fan_on एसी स्टार्ट
__label__fan_on एसी चलाओ
__label__fan_on एसी लगाओ
__label__fan_on फरहान ऑन
__label__fan_on फरहान चालू
__label__fan_on फरहान जलाओ
__label__fan_on फरहान स्टार्ट
__label__fan_on फरहान चलाओ
__label__fan_on फरहान लगाओ
__label__fan_on फैन चालू करो
__label__fan_on पंखा चला दो
__label__fan_on गर्मी लग रही है फैन ऑन करो
__label__fan_on स्विच ऑन द फैन
__label__fan_on हवा चाहिए पंखा चलाओ
```

EXAMPLE DATASET

5 Commands

Intent	Function	User Commands
wake_word	Activates the assistant and enables listening	**English:** Bharat **Hindi:** भारत
light_on	Turns the light ON	**English:** Light on, Turn on light **Hindi:** Light chalu karo, Batti jalao
light_off	Turns the light OFF	**English:** Light off, Turn off light **Hindi:** Light band karo, Batti bujhao
fan_on	Turns the fan ON	**English:** Fan on, Turn on fan **Hindi:** Pankha chalu karo, Hawa chalao
fan_off	Turns the fan OFF	**English:** Fan off, Turn off fan **Hindi:** Pankha band karo, Hawa roko
time	Tells the current system time	**English:** What is the time? **Hindi:** Time kya hai?, Kitne baje hain?
date	Tells today’s date	**English:** What is today’s date? **Hindi:** Aaj ki tareekh kya hai?
date (tomorrow)	Tells tomorrow’s date	**English:** Tomorrow’s date **Hindi:** Kal ki tareekh
math (addition)	Performs addition	**English:** 5 plus 3 **Hindi:** Paanch plus teen, Do jodo teen
math (subtraction)	Performs subtraction	**English:** 10 minus 2 **Hindi:** Das minus do, Das ghatao paanch
math (division)	Performs division	**English:** 20 divide 4 **Hindi:** Bees bhaag chaar
name_learning	Stores the user’s name	**Hindi:** Mera naam Rahul hai
ask_name	Recalls the stored user name	**Hindi:** Mera naam kya hai?
ask_identity	Tells the assistant’s identity	**English:** Who are you? **Hindi:** Tum kaun ho?
stop	Stops the assistant program	**English:** Stop **Hindi:** Ruk, Bas

6 Model Optimization

Model optimization was performed to enable efficient embedded deployment. Quantization techniques were applied to reduce model size and memory usage.

The optimized model provides:

- Reduced storage requirements
- Faster inference speed
- Lower computational overhead
- Improved real-time performance

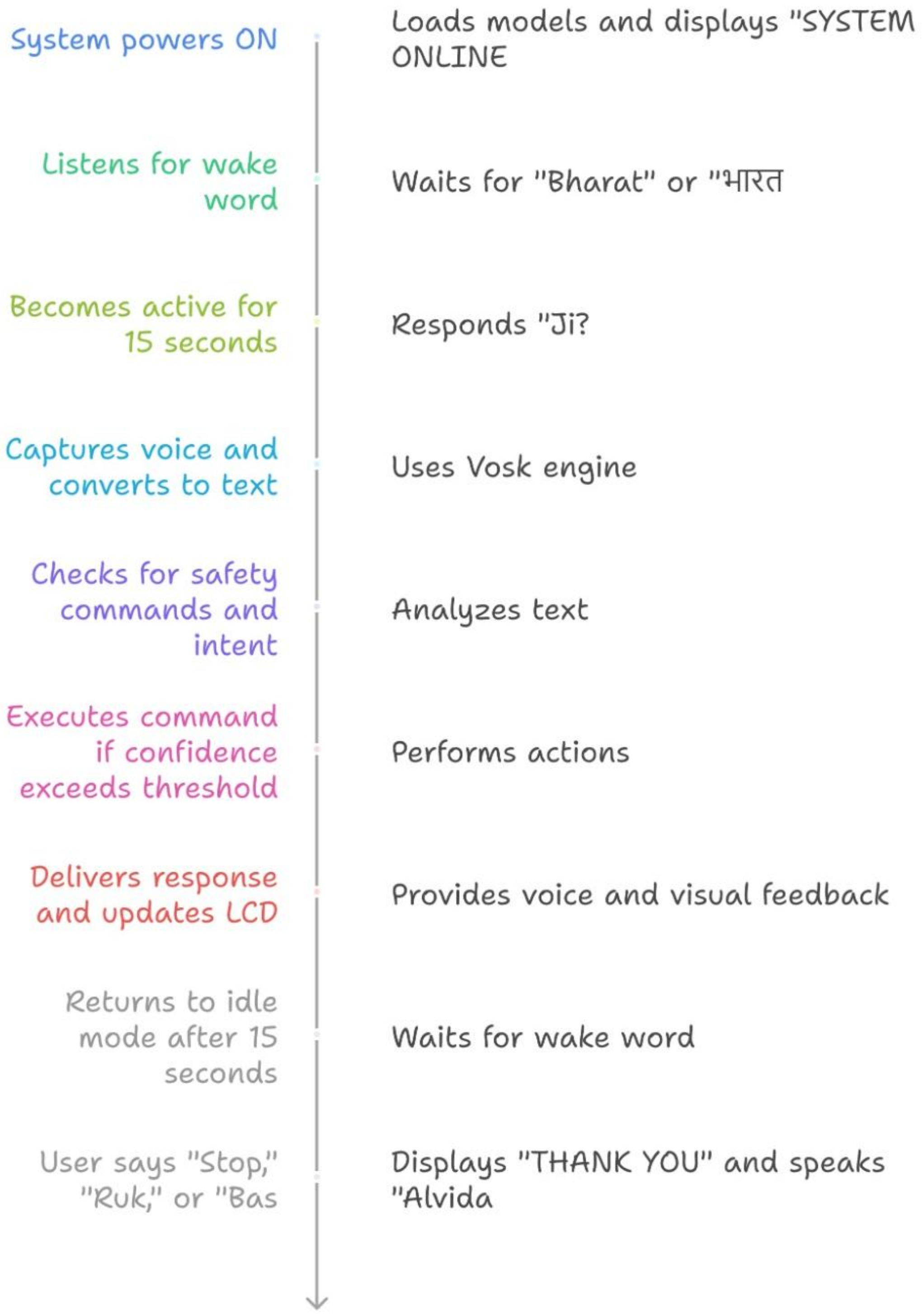
7 Methodology

The system operates through the following steps:

1. Wake word detection activates the assistant.(BHARAT)
2. Speech input is captured and converted to text.
3. FastText predicts user intent.
4. Fuzzy matching improves recognition accuracy.
5. Commands are executed and results displayed.

Additional Features

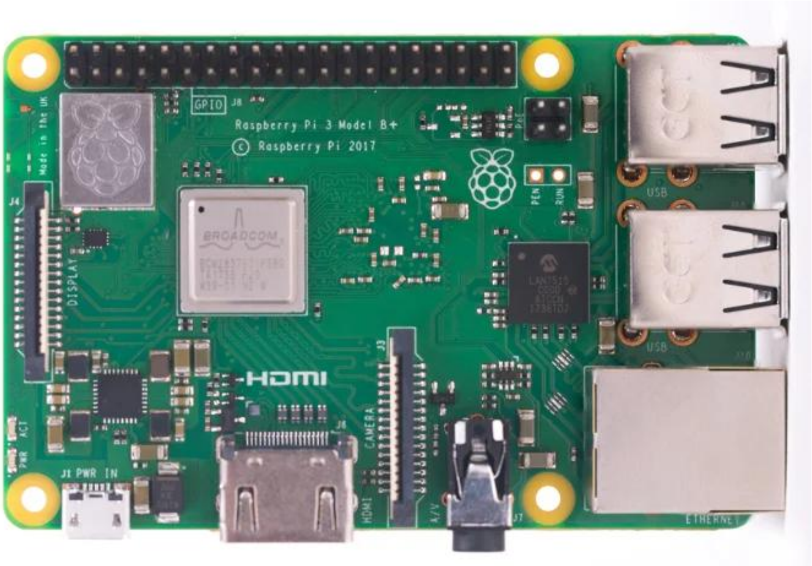
- Personalized user memory storage
- Mathematical computation module
- Device control simulation
- LCD feedback display


System powers ON
Loads models and displays "SYSTEM ONLINE
Listens for wake word
Waits for "Bharat" or "भारत
Becomes active for 15 seconds
Responds "Ji?
Captures voice and converts to text
Uses Vosk engine
Checks for safety commands and intent
Analyzes text
Executes command if confidence exceeds threshold
Performs actions
Delivers response and updates LCD
Provides voice and visual feedback
Returns to idle mode after 15 seconds
Waits for wake word
User says "Stop," "Ruk," or "Bas
Displays "THANK YOU" and speaks "Alvida

8 Hardware Utilization

The implementation uses Raspberry Pi as the primary processing unit. A microphone captures audio input, and an LCD display connected via I2C provides visual output. The system performs all computations locally without internet connectivity, ensuring privacy and reducing latency

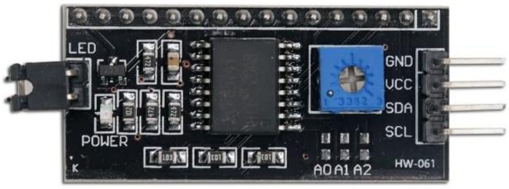
Raspberry Pi 4:



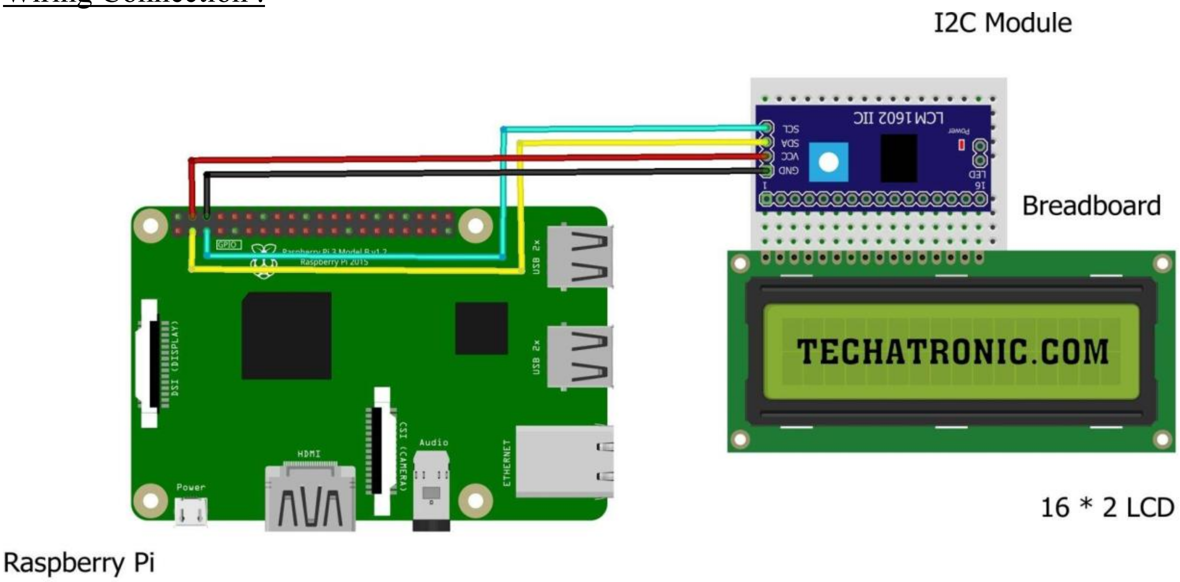
Microphone :

Spreaker (audio output):

LCD with I2C (visual output):



Wiring Connection :

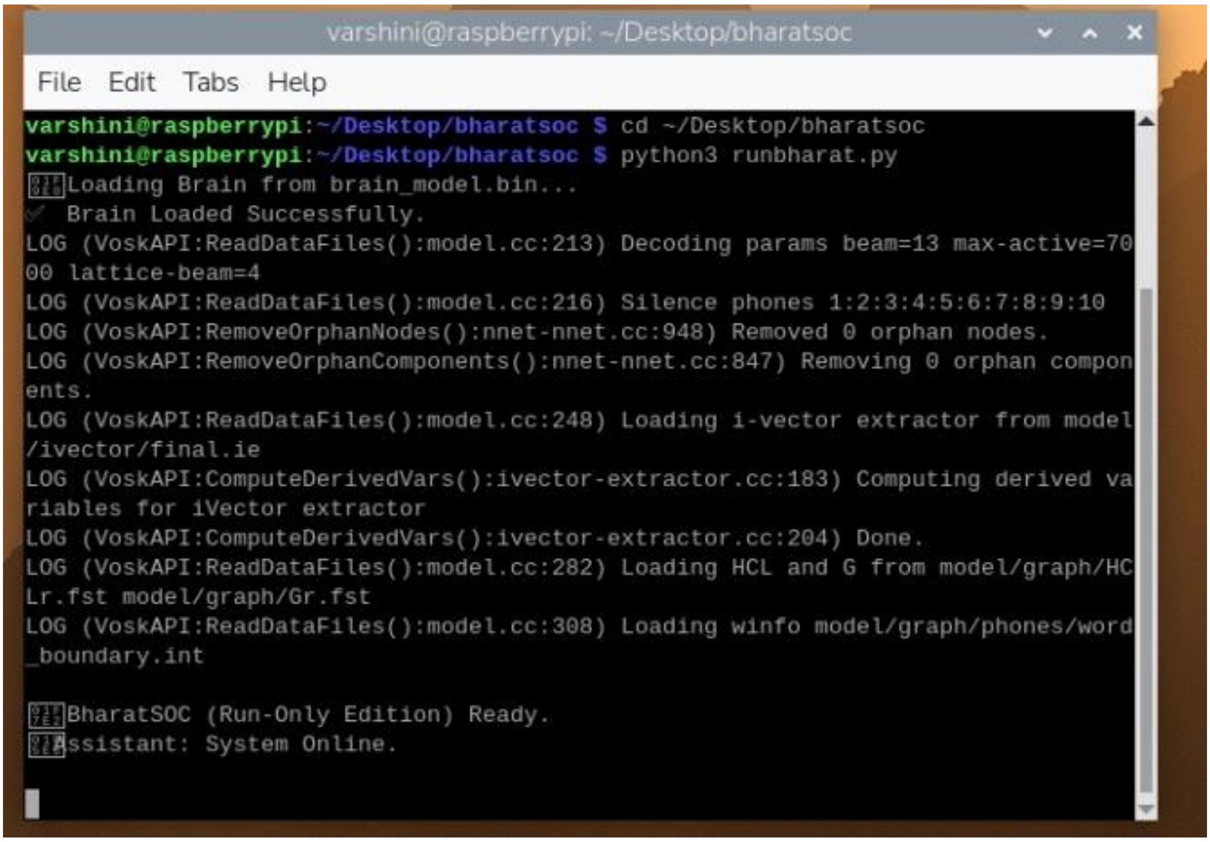


9 Results and Evaluation

The system was tested using multiple voice commands to evaluate performance. The assistant successfully recognized commands related to device control, mathematical operations, and information queries under minimal background noise.

The system demonstrated low latency response due to local processing. Intent classification accuracy was high due to labeled dataset training and fuzzy matching techniques.

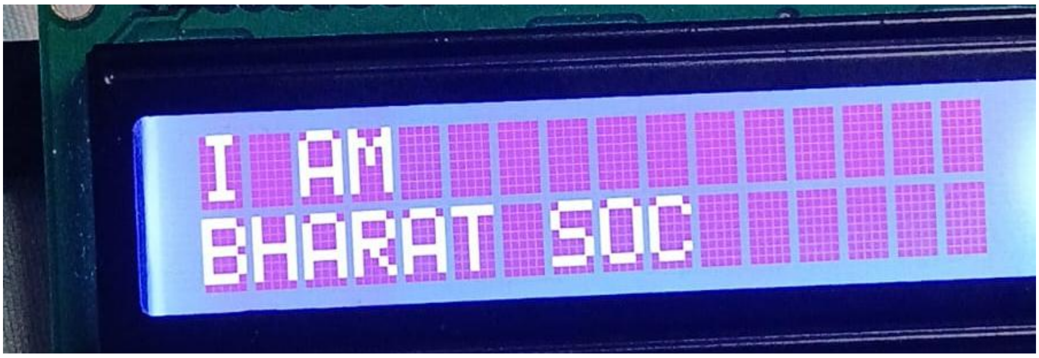
The optimized model reduced memory usage and improved execution speed .



```
varshini@raspberrypi:~/Desktop/bharatsoc $ cd ~/Desktop/bharatsoc
varshini@raspberrypi:~/Desktop/bharatsoc $ python3 runbharat.py
□Loading Brain from brain_model.bin...
✓ Brain Loaded Successfully.
LOG (VoskAPI:ReadDataFiles():model.cc:213) Decoding params beam=13 max-active=70
00 lattice-beam=4
LOG (VoskAPI:ReadDataFiles():model.cc:216) Silence phones 1:2:3:4:5:6:7:8:9:10
LOG (VoskAPI:RemoveOrphanNodes():nnet-nnet.cc:948) Removed 0 orphan nodes.
LOG (VoskAPI:RemoveOrphanComponents():nnet-nnet.cc:847) Removing 0 orphan compon
ents.
LOG (VoskAPI:ReadDataFiles():model.cc:248) Loading i-vector extractor from model
/ivector/final.ie
LOG (VoskAPI:ComputeDerivedVars():ivector-extractor.cc:183) Computing derived va
riables for iVector extractor
LOG (VoskAPI:ComputeDerivedVars():ivector-extractor.cc:204) Done.
LOG (VoskAPI:ReadDataFiles():model.cc:282) Loading HCL and G from model/graph/HC
Lr.fst model/graph/Gr.fst
LOG (VoskAPI:ReadDataFiles():model.cc:308) Loading winfo model/graph/phones/word
_boundary.int

□BharatSOC (Run-Only Edition) Ready.
□Assistant: System Online.
```

LCD OUTPUT IMAGE :



10 Challenges and Limitations

The project faced challenges such as handling background noise, dataset preparation, and limited computational resources. Speech recognition accuracy may vary depending on pronunciation and environmental conditions. Hardware limitations restrict the complexity of machine learning models.

10 Conclusion and Future Work

This project demonstrates the design and implementation of an offline privacy-preserving Hindi voice assistant using Edge AI techniques. The system integrates speech recognition, intent classification, and embedded hardware interaction to perform intelligent tasks locally.

Future work may include support for additional languages, improved speech recognition accuracy, expanded device control capabilities, and integration with Internet of Things devices.

GitHub Repository

https://github.com/RamavarshiniN/Offline-Privacy-Preserving-Hindi-Voice-Assistant-on-Raspberry-Pi