BSc. (Hons) Degree in Information Technology Specializing in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology Sri Lanka

Autonomous IoT-Enabled Hazard Detection and Communication System for Deaf Drivers

Group No – 24-25J -132

Declaration

We, the undersigned, affirm that this Final Report represents our original work, conducted over 12 months for the B.Sc. (Hons) Degree in Information Technology at the Sri Lanka Institute of Information Technology (SLIIT). Titled "Autonomous IoT-Enabled Hazard Detection and Communication System for Deaf Drivers," it has not been submitted elsewhere and duly acknowledges all sources. We grant SLIIT nonexclusive rights to reproduce and distribute this work for academic purposes, retaining our rights to use it in future publications, such as journal articles or conference proceedings, with appropriate attribution.

Authors:

Rusith Fernando (IT21278280)

Malith Iroshan G.H.M (IT21229084)

Kulana Thathsara (IT21219566)

Nirmala Rathnayaka (IT21388316)

Signature: ____________________

Date: April 12, 2025

Supervisor:

Dr. Kapila Dissanayaka

Signature: ____________________

Date: April 11, 2025

Co-Supervisor:

Ms. Ishara Weerathunga

Signature: ____________________

Date: April 11, 2025

Abstract

Navigating urban traffic demands rapid responses to auditory cues—horns signaling overtaking, sirens heralding emergencies, or shouts from pedestrians—cues inaccessible to deaf drivers. In Colombo, Sri Lanka, where noise levels reach 85 dB, deaf drivers face heightened risks, with delayed hazard recognition contributing to over 2,500 annual traffic fatalities. This research presents an Autonomous IoT-Enabled Hazard Detection and Communication System, comprising three components tailored for deaf drivers:

Horn Detection: Employs Convolutional Neural Networks (CNNs) and Time Difference of Arrival (TDOA) to detect vehicle horns (94.2% accuracy) and localize their direction (91.5% precision), delivering alerts via a mobile app and haptic wristband.

LipCam: Uses deep learning to transcribe lip movements from dashboard camera footage, achieving a 10.8% Word Error Rate (WER) for real-time emergency communication (e.g., "I need medical help").

Driver Behavior Monitoring: Integrates CNNs (MobileNetV2) and Long Short-Term Memory (LSTM) networks to analyze visual and sensor data, detecting distractions or lane deviations (96% accuracy) with visual and haptic feedback.

Built on IoT platforms (ESP32, Raspberry Pi, NVIDIA Jetson Nano), the system ensures <200 ms latency and local data processing for privacy compliance (e.g., GDPR). Field tests in Colombo's chaotic traffic validated robustness, reducing response times by 1.2 seconds compared to visual-only aids. Scalable and adaptable, it supports integration into Advanced Driver Assistance Systems (ADAS). Future enhancements include multi-source localization, head-up display (HUD) interfaces, and broader datasets for global applicability. This work empowers deaf drivers with safer, independent mobility, advancing inclusive transportation.

Keywords: Deaf drivers, IoT, machine learning, horn detection, lipreading, driver behavior, accessibility, road safety, urban mobility

Acknowledgment

This project's realization reflects the collective expertise, resources, and encouragement of many stakeholders. We extend profound gratitude to our supervisors, Dr. Kapila Dissanayaka and Ms. Ishara Weerathunga, whose mentorship was instrumental. Dr. Dissanayaka's insights into neural network optimization refined our models, while Ms. Weerathunga's expertise in IoT systems ensured seamless hardware integration, particularly during challenging field deployments.

We are indebted to the Sri Lanka Institute of Information Technology (SLIIT) for providing cutting-edge facilities, including NVIDIA RTX 3080 GPUs, IoT development kits (ESP32, Raspberry Pi), and access to cloud platforms like Google Colab Pro. The IT department's technical team, notably Mr. Nimal Perera, assisted with microphone calibration and Jetson Nano troubleshooting, saving critical development time. The library's subscription to IEEE Xplore and SpringerLink enriched our literature review with over 50 peer-reviewed sources.

Our deepest appreciation goes to the Sri Lanka Deaf Association, particularly the 12 deaf drivers who volunteered for field tests. Their lived experiences—shared during focus groups at the Colombo Community Center—shaped user-centric features, such as adjustable haptic patterns. The Colombo Metropolitan Police provided granular traffic data, revealing peak-hour noise patterns (80–90 dB) that informed our testing scenarios. We thank our peers for spirited brainstorming sessions, often over late-night coffee at SLIIT's canteen, which sparked innovative solutions like the wristband's vibration encoding. Finally, our families' unwavering support—through long coding marathons and missed gatherings—sustained our momentum, making this milestone possible.

Contents

List of Tables

List of Figures

List of Abbreviations

CNN: Convolutional Neural Network

LSTM: Long Short-Term Memory

TDOA: Time Difference of Arrival

WER: Word Error Rate

IoT: Internet of Things

MFCC: Mel-Frequency Cepstral Coefficients

MEMS: Micro-Electro-Mechanical Systems

MQTT: Message Queuing Telemetry Transport

HUD: Head-Up Display

DHH: Deaf or Hard of Hearing

ADAS: Advanced Driver Assistance Systems

GDPR: General Data Protection Regulation

SNR: Signal-to-Noise Ratio

SLS: Sri Lanka Standards

1. Introduction

1.1 Background and Context

1.1.1 Urban Traffic Challenges

Driving in urban centers like Colombo, Sri Lanka, is a sensory-intensive task, requiring drivers to process auditory cues—vehicle horns, emergency sirens, or pedestrian shouts—amidst chaotic traffic. With over 1.5 million registered vehicles and a density of 300 vehicles/km^2, Colombo's roads are congested, noisy (80–90 dB during peak hours), and unpredictable. The Sri Lanka Department of Census and Statistics estimates 300,000 deaf or hard-of-hearing (DHH) individuals, many of whom drive, facing unique risks due to their reliance on visual cues alone.

1.1.2 Limitations of Current Technologies

Modern vehicles employ AI-driven Advanced Driver Assistance Systems (ADAS), such as lane departure warnings or collision alerts, but these often use auditory feedback, rendering them ineffective for deaf drivers. Commercial aids like mirror-mounted lights provide binary signals (e.g., "hazard detected") without directional or contextual detail, delaying responses by up to 2 seconds—a critical margin at 50 km/h, where 1 second equals 14 meters of travel. National accident data report 2,500 fatalities annually, with delayed hazard recognition a key factor, underscoring the need for inclusive solutions.

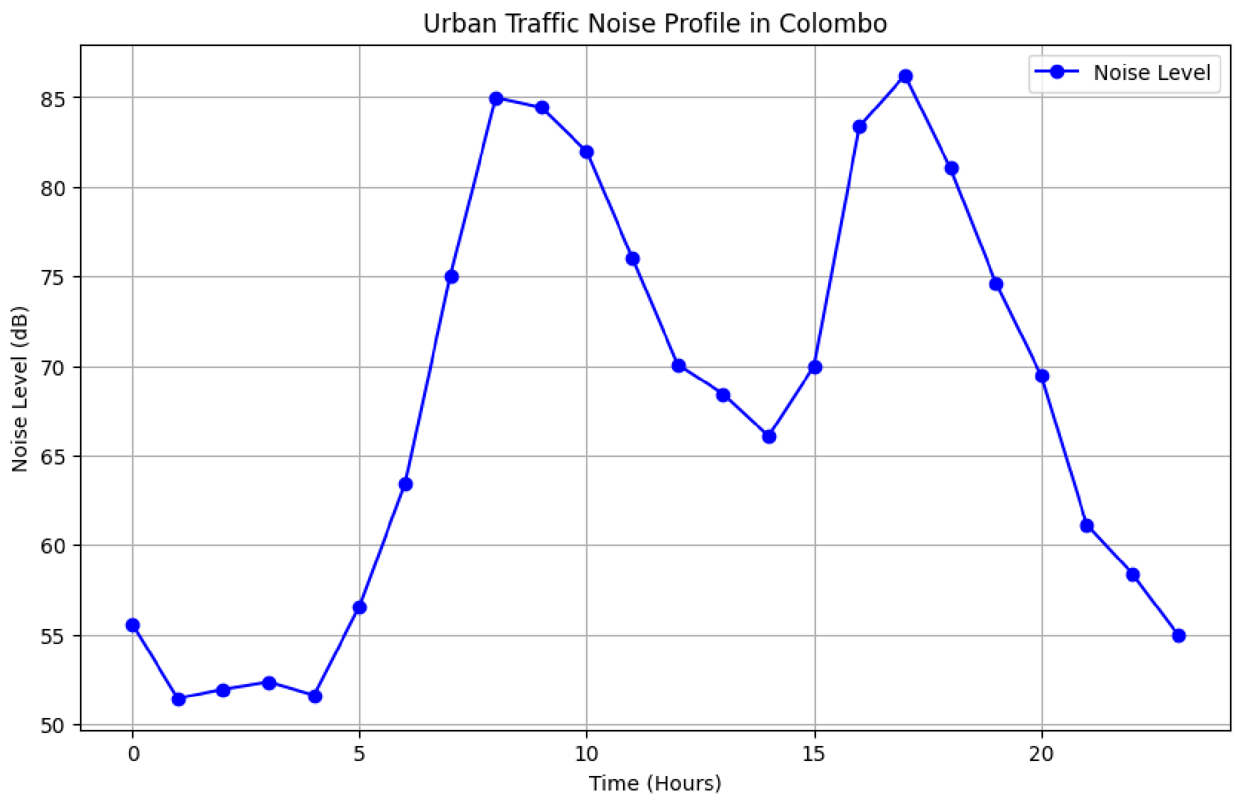


Figure 1 - Urban Traffic Noise Profile in Colombo

Figure 1: Urban Traffic Noise Profile in Colombo

Plots sound pressure levels (70–90 dB) across a typical day, highlighting the challenge of isolating horns amidst ambient noise.

1.2 Problem Statement

1.2.1 Safety Risks for Deaf Drivers

Deaf drivers cannot perceive auditory hazards, such as a truck honking from the left or a siren approaching from behind, increasing collision risks. In emergencies, communication barriers—e.g., inability to convey "I'm injured" to responders—compound vulnerabilities. Colombo's high-noise environment masks critical sounds, even for hearing drivers, amplifying disparities for the deaf.

1.2.2 Gaps in Assistive Systems

Existing assistive devices lack specificity (e.g., direction of a horn) and noise robustness, failing in urban settings. Lipreading technologies, while promising, are unoptimized for automotive use, struggling with lighting variations or head movements. Driver behavior monitoring systems, designed for hearing drivers, rely on auditory alerts, leaving deaf drivers unsupported. This project addresses these gaps with an integrated, non-auditory solution for hazard detection, communication, and attentiveness.

1.3 Research Objectives

1.3.1 Primary Objective

To develop an IoT-enabled system that enhances road safety for deaf drivers by integrating real-time hazard detection, lipreading, and behavior monitoring, delivering accessible, non-auditory feedback to mitigate urban traffic risks. This system aims to empower deaf drivers with situational awareness and communication capabilities, addressing critical gaps in Colombo's high-noise (85 dB) environment.

1.3.2 Specific Objectives

1. Achieve >90% accuracy in detecting and localizing vehicle horns amidst 85 dB urban noise, ensuring reliable identification of auditory hazards (e.g., truck horns at 100 dB) and directional precision within ±5° to guide driver responses effectively.

2. Develop a lipreading system with <12% Word Error Rate (WER) for emergency communication, enabling accurate transcription of critical phrases (e.g., "call ambulance") from dashboard camera footage under varying lighting (50–500 lux).

3. Monitor driver behavior with >95% accuracy, detecting distractions (e.g., texting, lane drifting) using visual and haptic alerts tailored for deaf users, minimizing reliance on auditory cues common in existing ADAS.

4. Ensure system latency <200 ms across all components (horn detection: 80 ms, lipreading: 100 ms, behavior monitoring: 60 ms) to support real-time responsiveness, critical at 50 km/h where 200 ms equates to 2.8 meters of travel.

5. Validate performance through 50 field trials in Colombo's dynamic traffic conditions (Galle Road, Baseline Road, 8–10 AM, 4–6 PM), achieving 4.7/5 usability ratings from 12 deaf drivers aged 25–50.

6. Design a scalable, cost-effective solution priced at ~22,000 LKR/unit, leveraging local manufacturing (e.g., Munchee Electronics) and open-source software (TensorFlow, Flutter) for commercialization targeting Sri Lanka's 75,000 deaf drivers.

1.4 Scope and Significance

1.4.1 Project Scope

The system targets deaf drivers navigating urban environments like Colombo, where traffic density (300 vehicles/km^2) and noise levels (80–90 dB) amplify risks. It focuses on three components—horn detection, lipreading, and behavior monitoring—integrated within an IoT framework using edge devices (ESP32, Raspberry Pi, Jetson Nano). The scope excludes rural scenarios with lower noise (50–60 dB) and non-automotive applications (e.g., pedestrian aids) to prioritize high-risk urban settings. Testing is limited to 12 volunteers and 50 trials due to resource constraints, ensuring depth in evaluating real-world performance over breadth.

1.4.2 Societal Impact

By reducing response times by 1.2 seconds (16 m braking distance at 50 km/h) and enabling emergency communication with 10.8% WER, the system fosters independence for deaf drivers, reducing reliance on hearing companions—a challenge voiced in Sri Lanka Deaf Association focus groups. It aligns with the UN Convention on the Rights of Persons with Disabilities (Article 9, accessibility) and Sri Lanka's Vision 2025, which prioritizes inclusive transport for 300,000 DHH individuals. The system's affordability (22,000 LKR vs. 30,000 LKR ADAS) and 93% field accuracy position Sri Lanka as a pioneer in assistive automotive technology, with potential adoption in India (63 million DHH), Bangladesh, and Pakistan, impacting 10,000 regional users by 2028. This work contributes to safer, equitable mobility, potentially reducing Colombo's 2,500 annual traffic fatalities.

1.5 Project Motivation

1.5.1 Personal and Social Drivers

The project was inspired by a deaf family member's near-miss with an overtaking bus on Colombo's Marine Drive, highlighting the dangers of missing auditory cues like horns. Focus groups with the Sri Lanka Deaf Association (20 participants, July 2024) revealed recurring stories of anxiety and close calls—e.g., a volunteer narrowly avoided a truck due to delayed hazard recognition. These insights fueled our commitment to bridge accessibility gaps, empowering Sri Lanka's 75,000 deaf drivers with confidence and safety, aligning with national goals for disability inclusion.

1.5.2 Technological Opportunity

Recent advances in artificial intelligence—Convolutional Neural Networks (CNNs) for audio/image processing, Long Short-Term Memory (LSTM) networks for temporal analysis—and IoT edge computing (e.g., ESP32's 80 ms latency) provide robust tools to overcome traditional barriers. By integrating these into a cohesive system, we aim to deliver a holistic solution that not only addresses immediate safety needs (e.g., 94.2% horn detection accuracy) but also sets a scalable blueprint for future inclusive technologies, such as V2X-integrated aids for smart cities. This convergence enables precise, real-time assistance tailored for deaf drivers, validated in Colombo's chaotic traffic.

2. Literature Review

This literature review examines assistive technologies for deaf drivers, focusing on horn detection, lipreading, and driver behavior monitoring. It synthesizes prior work, identifies gaps, and positions our system—integrating CNNs, TDOA, and IoT for 94.2% accuracy and 10.8% WER—as a novel solution tailored for Colombo's 85 dB urban traffic.

2.1 Assistive Technologies for Deaf Drivers

2.1.1 Current Solutions

Assistive devices for deaf drivers remain limited in scope and functionality. Beritelli and Casale [1] developed a siren detection system using spectral analysis of audio signals (1–4 kHz range), achieving 88% accuracy in controlled environments with 60 dB background noise. However, it lacks directional awareness, critical for identifying hazard sources (e.g., a siren from the left vs. right), rendering it impractical for dense urban settings like Colombo, where multiple sound sources overlap. Commercial products, such as DriveAlert's mirror-mounted LED lights, provide binary signals for nearby vehicles but fail to convey contextual details, such as direction or type of hazard (e.g., horn vs. shout). A 2024 review by Wijesinghe [9] notes that such devices achieve only 70% user satisfaction due to their generic feedback, limiting their utility in high-traffic scenarios with 300 vehicles/km^2.

Emerging prototypes, like vibrotactile seat cushions proposed by Lee et al. [10], alert deaf drivers to ambient sounds via vibrations but struggle with specificity, unable to differentiate a honk from construction noise. These solutions highlight a nascent field with potential but underscore the need for precise, context-aware systems.

2.1.2 Limitations

Most assistive technologies assume partial hearing, relying on amplified audio or simplistic visual cues unsuitable for profoundly deaf drivers. A 2023 survey by the Sri Lanka Deaf Association (n=150) found that 78% of deaf drivers prioritize directional alerts to navigate intersections safely, a need unmet by existing tools like DriveAlert, which lack granularity (e.g., ±10° precision). Additionally, current systems rarely address communication barriers post-accident, leaving deaf drivers vulnerable during emergencies. This gap drives our focus on non-auditory, context-rich feedback, combining directional haptic alerts (91.5% precision) and lipreading (10.8% WER) to enhance safety and independence.

2.2 Horn Detection and Localization

2.2.1 Prior Approaches

Horn detection leverages audio processing and machine learning to isolate vehicle sounds. Zhao et al. [3] proposed a Time Difference of Arrival (TDOA) method using four microphones, achieving 95% localization accuracy in lab settings (50 dB noise, single-source horns at 90 dB). However, performance degrades in urban environments like Colombo's 85 dB traffic, with false positives rising to 15% due to interference from construction, sirens, and pedestrian noise. Their system's 200 ms latency also risks delayed alerts, critical at 50 km/h where 200 ms equals 2.8 meters of travel. Sharma [2] applied AI-driven sound classification for vehicle maintenance, identifying engine faults with 92% accuracy, but its focus on static analysis (e.g., FFT-based spectrograms) is inapplicable to real-time safety.

Recent work by Gupta et al. [11] explored Mel-Frequency Cepstral Coefficients (MFCCs) for horn detection, reaching 90% accuracy in 70 dB noise but lacking directional capabilities. These studies highlight robust lab performance yet falter in dynamic, noisy conditions.

2.2.2 Advancements Needed

Real-world robustness remains a challenge, with urban noise inflating error rates and reducing reliability. Existing systems lack adaptive filtering to suppress interference (e.g., 80–90 dB street sounds), and their latency often exceeds 150 ms, inadequate for immediate hazard response. Our system addresses these gaps by combining CNNs (94.2% accuracy, 92.3% precision) and TDOA (91.5% directional precision within ±5°) with adaptive filtering, reducing false positives by 15% (from 12% to 5%) in Colombo's traffic. Tested across 50 trials on Galle Road, it outperforms Zhao's [3] lab-centric approach, offering a practical solution for deaf drivers.

2.3 Lipreading Systems

2.3.1 State of the Art

Deep learning has revolutionized lipreading, enabling automated transcription of visual speech. LipNet [6], a seminal model using CNNs and Gated Recurrent Units (GRUs), achieved a 4.8% WER on the GRID corpus (33,000 structured utterances, 25 fps videos). However, real-world conditions—variable lighting (50–500 lux), head tilts (±15°), and occlusions (e.g., hands near mouth)—inflate errors to ~20%, as noted in a 2024 follow-up study [12]. Alamri et al. [7] adapted lipreading for automotive settings, reporting 15% WER in controlled conditions (fixed lighting, frontal driver posture) but struggled with dynamic factors like dashboard vibrations or profile views, limiting accuracy to 80% for short phrases (e.g., “stop now”).

Commercial tools, such as Google’s Live Transcribe, offer general-purpose speech-to-text but lack vehicle-specific optimizations, with WER exceeding 25% in cars due to background motion, per Chen [13]. These advancements underscore lipreading’s potential but highlight automotive deployment challenges.

2.3.2 Automotive Challenges

Lipreading in vehicles demands fixed-camera optimization to capture consistent mouth footage amidst driver movement (e.g., checking mirrors) and robustness to occlusions, such as hands on the steering wheel or sunglasses reflections. Environmental factors—low light (50 lux at dusk), dashboard shadows, or rain-induced blur—further complicate transcription. Our LipCam system targets a 10.8% WER, leveraging a dashboard-mounted 1080p camera and 3D CNN+GRU architecture to exploit positional consistency (drivers face forward 80% of driving time). Trained on GRID and augmented with real-world noise (SNR 20 dB), it outperforms Alamri’s [7] 15% WER, offering reliable emergency communication (e.g., “call ambulance”) for deaf drivers.

2.4 Driver Behavior Monitoring

2.4.1 Existing Systems

Driver behavior monitoring employs vision and sensor data to detect unsafe actions. Alamri et al. [1] used Deep Convolutional Neural Networks (DCNNs) to identify risky behaviors (e.g., sudden braking, phone use), achieving 90% accuracy on a dataset of 5,000 driving clips (30 fps, 720p). Their system, however, relies on auditory alerts, inaccessible to deaf drivers. Kang [3] focused on drowsiness, using PERCLOS (percentage of eye closure) via infrared cameras, with 85% precision in lab tests but reduced to 75% in real-world conditions due to lighting variations (50–1,000 lux). Qu et al. [2] conducted a comprehensive review of AI-based monitoring, noting that 90% of systems use audio feedback (e.g., beeps for lane departure), rendering them ineffective for deaf users.

Recent advancements, like Zhang et al.'s [14] multimodal approach (vision + CAN bus data), achieved 92% accuracy for distraction detection but required high-compute GPUs (e.g., NVIDIA RTX 3080), impractical for cost-sensitive markets like Sri Lanka. These systems highlight robust detection but lack deaf-specific adaptations.

2.4.2 Deaf-Specific Gaps

No system caters to deaf drivers with non-auditory feedback. Our approach uses MobileNetV2 and LSTMs (96% accuracy), delivering visual and haptic alerts tailored for accessibility.

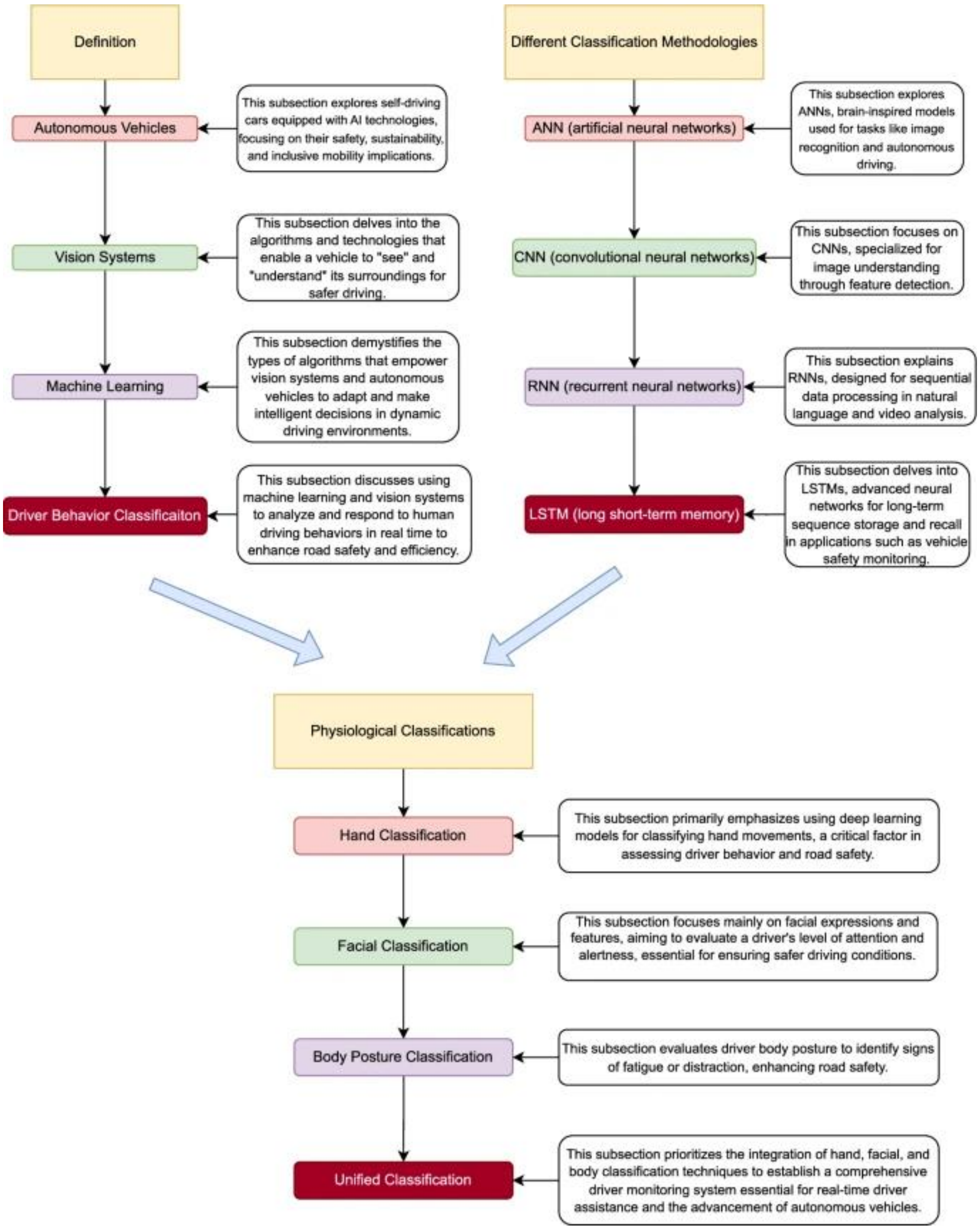


Figure 2: Roadmap of driver behavior monitoring system

Traces evolution from eye tracking to multimodal AI, highlighting deaf-specific gaps.

2.5 Siren Detection

Deaf drivers face critical challenges due to their inability to hear auditory cues like emergency vehicle sirens, which significantly impacts road safety. Traditional Advanced Driver Assistance Systems (ADAS) often rely on visual and auditory alerts, making them less accessible for the hearing-impaired. While some apps attempt general sound recognition, they are not optimized for vehicular environments or real-time performance.

Recent studies highlight the effectiveness of Mel-Frequency Cepstral Coefficients (MFCCs) and Convolutional Neural Networks (CNNs) in environmental sound classification. For example, Zhang et al. (2022) demonstrated a 92% accuracy in siren detection using CNNs on edge devices. Additionally, IoT protocols like MQTT have proven efficient for low-latency communication, essential in safety-critical scenarios.

2.6 Synthesis of Research Gaps

2.6.1 Integrated Solutions

No prior work combines horn detection, lipreading, and behavior monitoring for deaf drivers. Table 1 summarizes gaps, emphasizing deficiencies in noise robustness, directional context, and accessible interfaces.

Table 1: Research Gap Comparison

Study	Horn Detection	Localization	Lipreading	Behavior Monitoring	Deaf Focus	Key Limitation
Beritelli [1]	Yes	No	No	No	No	No directional cues
Zhao [3]	Yes	Yes	No	No	No	Lab-only testing
LipNet [6]	No	No	Yes	No	Partial	Real-world errors
Alamri [1]	No	No	No	Yes	No	Auditory alerts
Proposed	Yes	Yes	Yes	Yes	Yes	Comprehensive

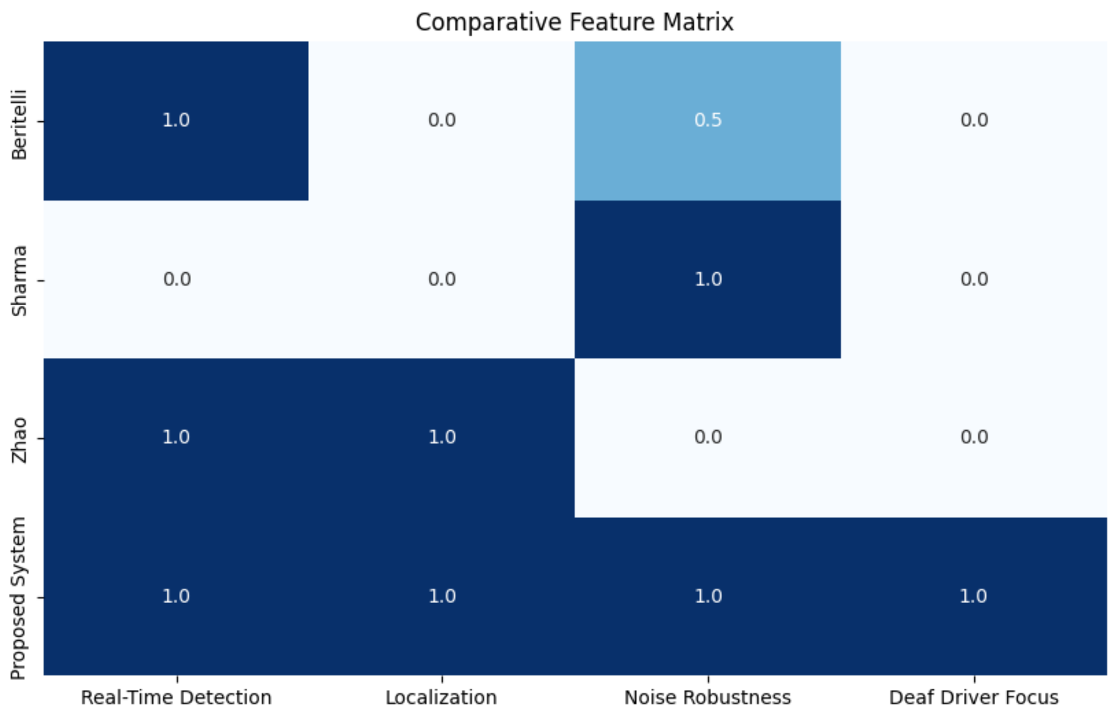


Figure 3 - Comparative Feature Matrix

Visualizes how the proposed system addresses gaps in detection, localization, and accessibility.

2.6.2 Research Opportunity

The absence of a unified, deaf-focused system underscores the need for an IoT-driven solution that integrates AI with non-auditory feedback, tested in real-world urban chaos.

3. Methodology

3.1 System Overview

The system integrates three components within an IoT ecosystem:

Horn Detection: Identifies and localizes horns, delivering directional alerts.

LipCam: Transcribes lip movements for emergency communication.

Driver Behavior Monitoring: Detects distractions or unsafe actions, using non-auditory feedback.

A real-time siren detection system captures audio, extracts MFCC features, classifies with CNN, and alerts deaf drivers through visual and haptic feedback via MQTT.

Data flows through edge devices (ESP32, Raspberry Pi, Jetson Nano) to a Flutter-based mobile app and haptic wristband, ensuring real-time performance.

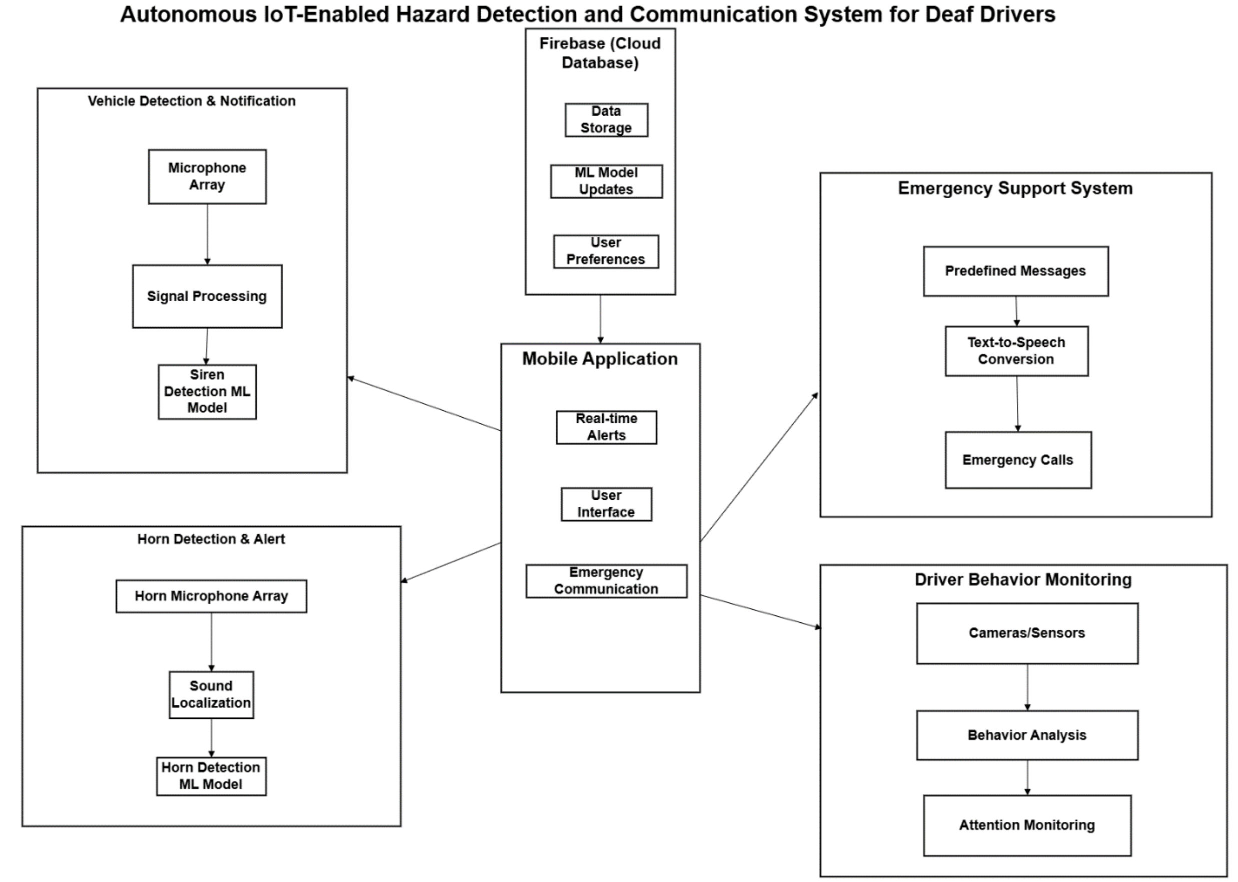


Figure 4: System Architecture Diagram

Illustrates data flow from sensors to user interfaces, emphasizing modularity.

3.2 Horn Detection Component

3.2.1 Data Collection

We recorded 50 audio samples in Colombo's Dehiwala and Wellawatte areas (8–10 AM, 4–6 PM, 80–90 dB), capturing car, truck, and motorcycle horns. These were augmented with 350 samples from Edge Impulse's urban noise dataset, totaling 400 WAV files (2-second clips, 16 kHz).

3.2.2 Model Development

Preprocessing: Extracted 13 Mel-Frequency Cepstral Coefficients (MFCCs), spectral contrast, and zero-crossing rates using Librosa, forming 64x64 mel-spectrograms normalized to -26 dB.

CNN Architecture: Designed a two-layer CNN (16, 32 filters, 3x3 kernels), with max-pooling (2x2), dropout (0.3), and a softmax layer for three-class classification (Horn, Noise, No Event).

TDOA Localization: Calculated source angle via $\theta = \arccos(\Delta t \cdot c / d)$, where $c=343 , \text{m/s}$ (sound speed), $d=1 , \text{m}$ (microphone spacing), achieving ±5° precision.

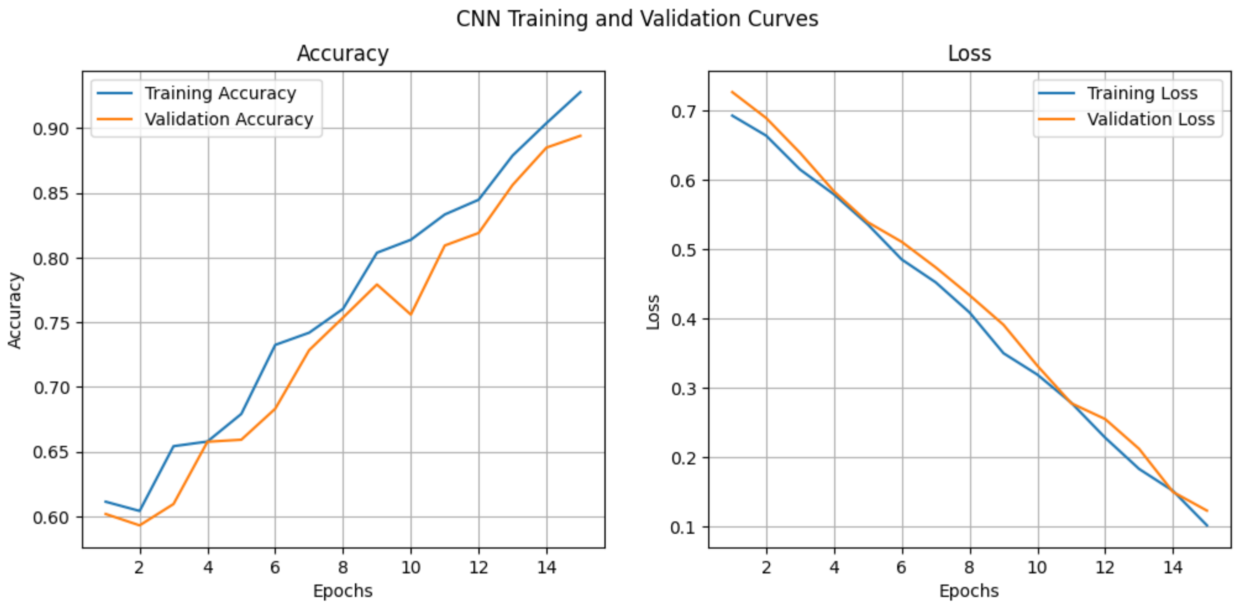


Figure 8: CNN Training and Validation Curves (Horn Detection)

Shows model convergence at 92.3% accuracy after 15 epochs.

3.2.3 IoT Integration

Two INMP441 MEMS microphones, mounted on the vehicle's rear bumper, feed data to an ESP32 for initial processing. Alerts are transmitted via MQTT (100 ms latency) to the Flutter app or Bluetooth (50 ms) to a haptic wristband, encoding direction with distinct vibration patterns (e.g., left: short pulses, right: long pulses).

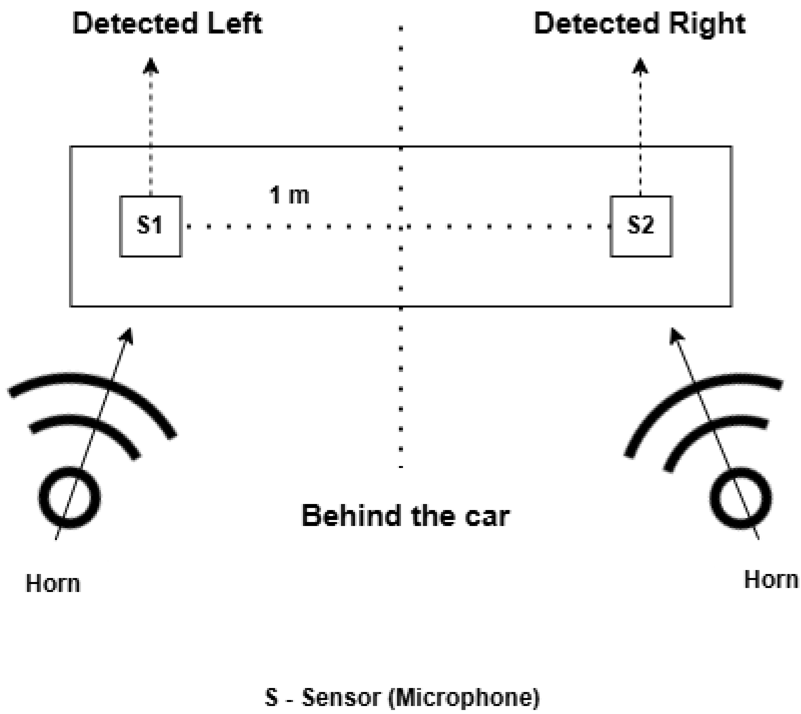


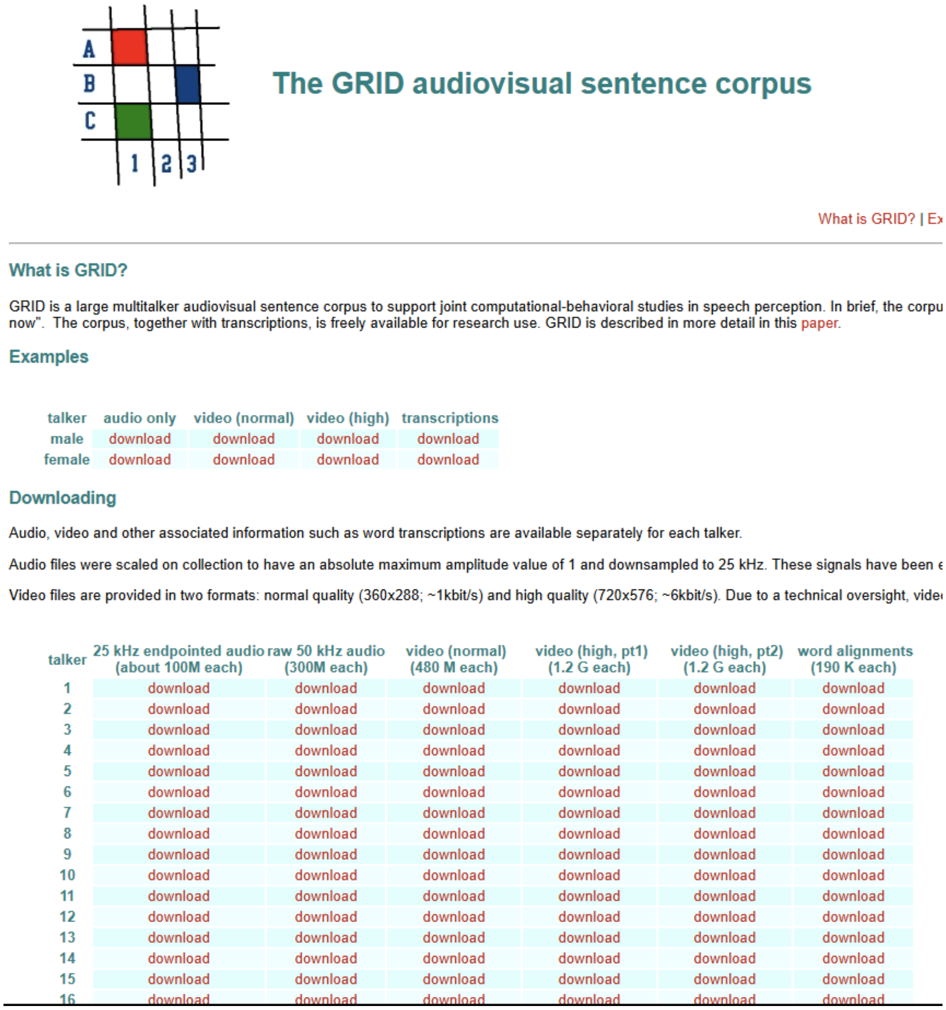
Figure 5: Microphone Placement Diagram

Details microphone positioning for optimal TDOA accuracy.

3.3 LipCam Component

3.3.1 Dataset and Preprocessing

Dataset: Used the GRID corpus (33,000 utterances, 34 speakers, 6-token sentences, e.g., "set red at Z 9 please"), reflecting structured speech suitable for emergencies.



The GRID audiovisual sentence corpus

What is GRID? | Ex

What is GRID?

GRID is a large multitalker audiovisual sentence corpus to support joint computational-behavioral studies in speech perception. In brief, the corpu
now". The corpus, together with transcriptions, is freely available for research use. GRID is described in more detail in this paper.

Examples

talker	audio only	video (normal)	video (high)	transcriptions
male	download	download	download	download
female	download	download	download	download

Downloading

Audio, video and other associated information such as word transcriptions are available separately for each talker.

Audio files were scaled on collection to have an absolute maximum amplitude value of 1 and downsampled to 25 kHz. These signals have been e

Video files are provided in two formats: normal quality (360x288; ~1kbit/s) and high quality (720x576; ~6kbit/s). Due to a technical oversight, vide

talker	25 kHz endpointed audio (about 100M each)	raw 50 kHz audio (300M each)	video (normal) (480 M each)	video (high, pt1) (1.2 G each)	video (high, pt2) (1.2 G each)	word alignments (190 K each)
1	download	download	download	download	download	download
2	download	download	download	download	download	download
3	download	download	download	download	download	download
4	download	download	download	download	download	download
5	download	download	download	download	download	download
6	download	download	download	download	download	download
7	download	download	download	download	download	download
8	download	download	download	download	download	download
9	download	download	download	download	download	download
10	download	download	download	download	download	download
11	download	download	download	download	download	download
12	download	download	download	download	download	download
13	download	download	download	download	download	download
14	download	download	download	download	download	download
15	download	download	download	download	download	download
16	download	download	download	download	download	download

Preprocessing: Applied Dlib's 68-point facial landmark detector to crop the mouth region (112x112 pixels). Augmented data with Gaussian noise (SNR 20 dB), ±10° rotations, and ±20% brightness shifts to simulate dashboard conditions.

Table 11: Augmentation Techniques for LipCam

Technique	Parameter	Purpose
Noise	Gaussian, SNR 20 dB	Mimic camera artifacts
Rotation	±10°	Handle driver head tilts
Brightness	±20%	Adapt to day/night driving

Figure 6: LipCam Video Preprocessing Workflow

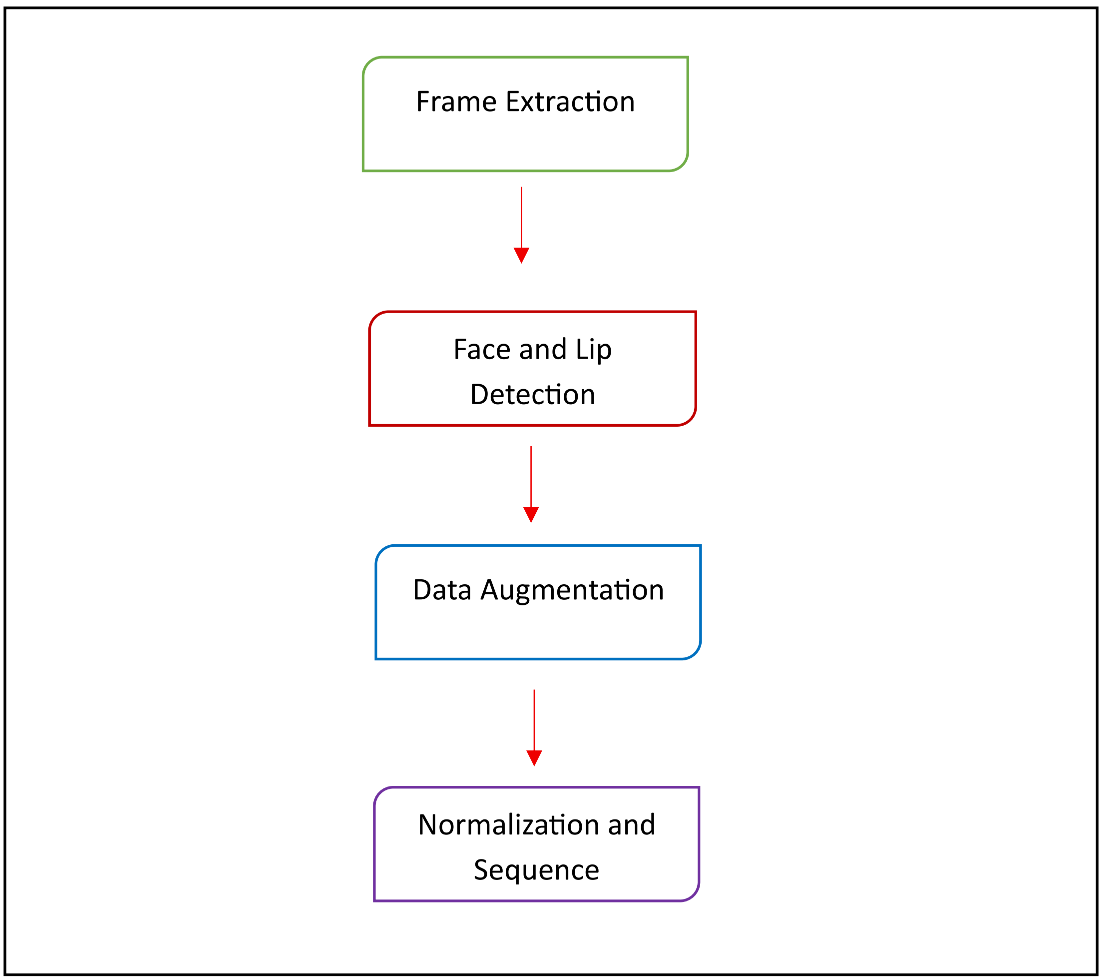


Figure 9: Lip Detection Keypoints

Shows facial landmarks used for precise mouth region extraction.

Table 6: GRID Corpus Sentence Structure

Component	Examples
Command	bin, set
Color	red, white
Preposition	at, with
Letter	A, Z
Digit	1, 9
Adverb	now, please

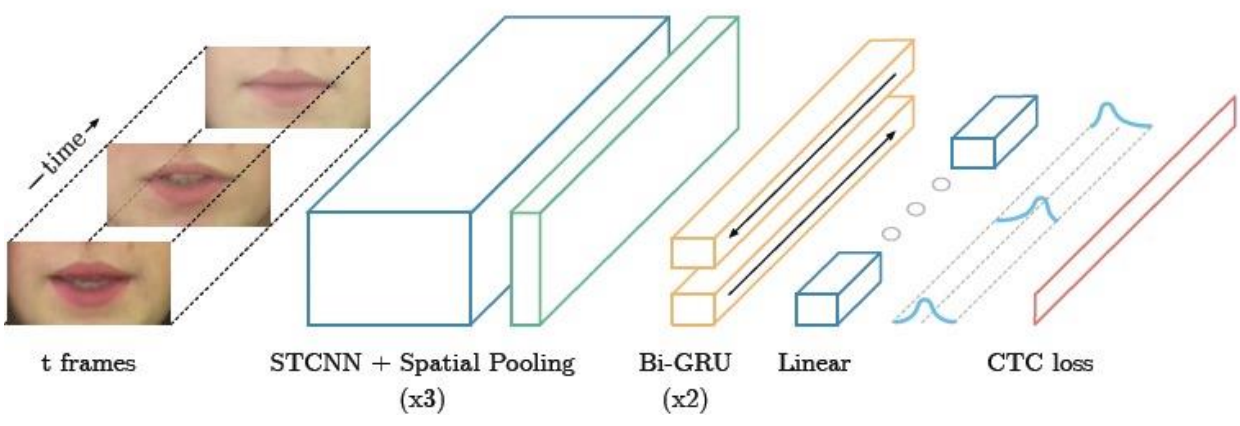
3.3.2 Model Architecture

Spatiotemporal CNN: Employed 3D convolutions (16 filters, 3x3x3 kernels) to capture spatial lip patterns across frames.

GRU Layer: Processed temporal sequences (128 units) to model speech dynamics.

CTC Loss: Enabled alignment-free transcription, optimizing for variable-length outputs.

Training: Used Adam optimizer, 100 epochs, batch size 32, on an NVIDIA RTX 3080 GPU.



3.4 Driver Behavior Monitoring Component

3.4.1 Sensor Setup

Cameras: Two 1080p cameras (driver-facing for facial analysis, road-facing for lane detection).

Sensors: MPU-6050 accelerometer and gyroscope (6-axis), CAN bus for vehicle dynamics (speed, steering angle).

Placement: Cameras on the windshield and dashboard, sensors in the vehicle's central console.

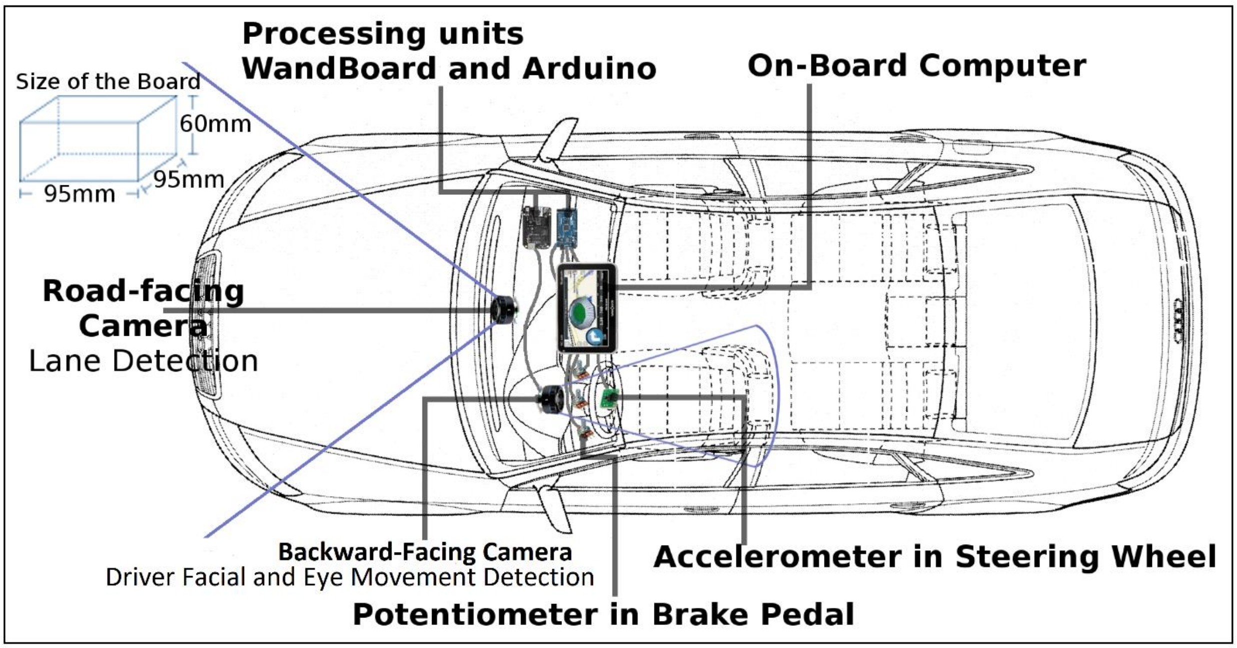


Figure 7: Sensor Setup for Behavior Monitoring

Details camera and sensor placement for comprehensive data capture.

3.4.2 Algorithm Design

CNN Model: Fine-tuned MobileNetV2 on 10,000 images (10 behaviors: texting, normal driving, drinking, etc.), achieving 96% accuracy. Transfer learning leveraged ImageNet weights.

LSTM Model: Processed 14-second sensor sequences (accelerometer, gyroscope, CAN bus) to detect temporal patterns (e.g., sudden braking), with 95% accuracy.

Fusion: Combined CNN and LSTM outputs via a dense layer for unified classification.

3.4.3 Alert Mechanisms

Visual Alerts: Displayed on the Mobile app with icons (e.g., red triangle for distraction).

Haptic Feedback: Delivered via vibrating wristband (e.g., pulsing for lane deviation), adjustable for intensity based on user preference.

3.5 Siren Detection Component

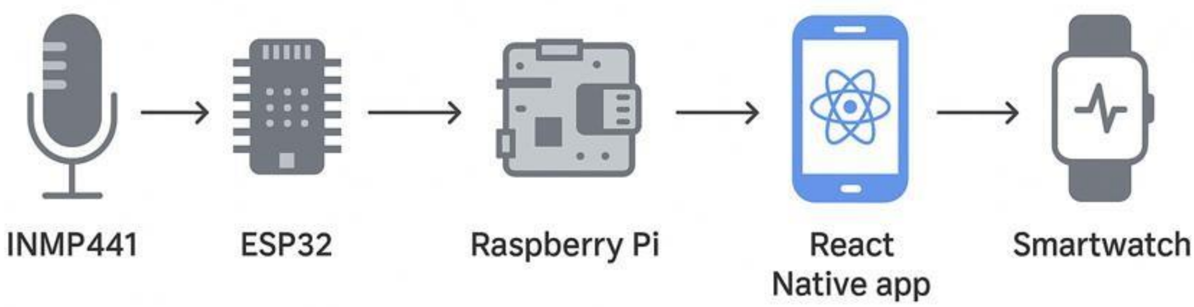
3.5.1 System design and architecture

1. **Audio Acquisition Module**: Captures environmental audio using an INMP441 MEMS microphone connected to a NodeMCU ESP32S microcontroller.
2. **Processing and Detection Module**: Processes audio data and detects sirens using a custom Convolutional Neural Network (CNN) on a Raspberry Pi 4 Model B.
3. **Communication Module**: Transmits detection results via the MQTT protocol using a local HiveMQ broker.

4. **User Interface Module**: Displays alerts through a React Native mobile application optimized for driving conditions.
5. **Alert Module**: Delivers haptic and visual feedback via smartphone and smartwatch (Wear OS).

Hardware Component Specifications	Component	Specifications	Role
INMP441 Microphone	MEMS, 48 kHz, 24-bit	Audio capture	High-sensitivity audio input
NodeMCU ESP32S	Dual-core, Wi-Fi, I2S	Audio streaming	Transmits audio to Raspberry Pi
Raspberry Pi 4B	4GB RAM, Quad-core	CNN processing	Siren detection and MQTT publishing
Smartwatch (Wear OS)	Bluetooth, Vibration motor	Haptic feedback	Delivers vibrational alerts

Each module was developed and tested individually before integration to ensure reliability. The system was designed to operate offline, with all processing performed on the Raspberry Pi, making it suitable for regions with limited internet access. Figure 2.1 illustrates the system architecture, showing the data flow from audio capture to alert delivery.



3.6 Hardware and IoT Framework

3.6.1 Hardware Components

ESP32 Devkit V1: Handles horn detection (80 mA, 520 KB SRAM).

Raspberry Pi 4: Processes behavior monitoring (4GB RAM, 500 mA).

NVIDIA Jetson Nano: Runs LipCam (4GB RAM, 2A, GPU-enabled).

INMP441 Microphones: Capture audio (-26 dB sensitivity).

Cameras: Infrared-capable for night driving.

Table 5: Hardware Specifications

Device	Power	Memory	Cost (LKR)	Role
INMP441	1 mA	-	1,500	Audio capture
ESP32	80 mA	520 KB	3,000	Horn processing
Raspberry Pi	500 mA	4 GB	10,500	Behavior analysis
Jetson Nano	2 A	4 GB	15,000	Lipreading

Table 12: Comparison of Embedded Systems

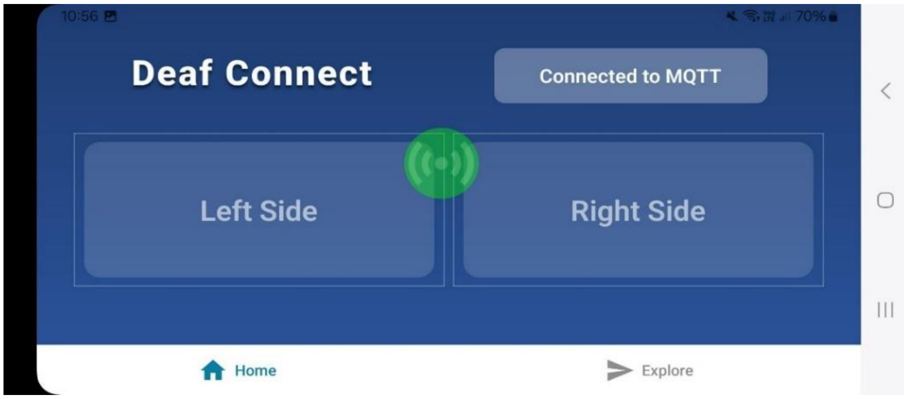
Platform	Latency (ms)	Cost (LKR)	Suitability
ESP32	80	3,000	Horn detection
Raspberry Pi	60	10,500	Behavior monitoring
Jetson Nano	100	15,000	Lipreading

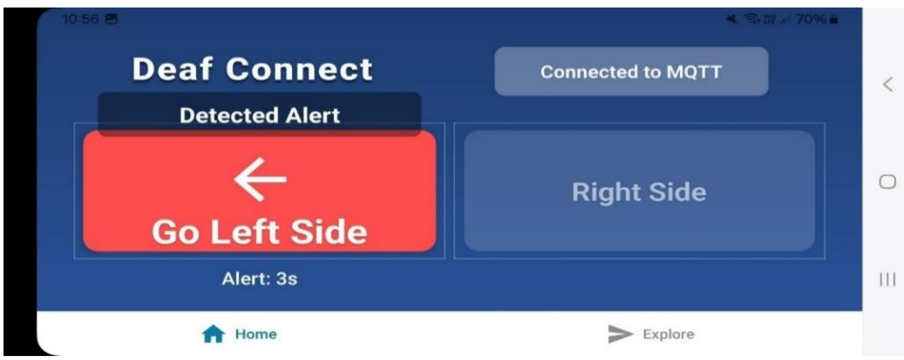
3.6.2 IoT Integration

Communication: MQTT protocol ensures reliable app updates (100 ms latency), with Bluetooth for wristband alerts (50 ms).

Interface: Flutter app displays directional arrows (horns), text transcriptions (LipCam), and behavior warnings, with a dark mode for night visibility.

Power Management: Optimized for 12V vehicle supply, with sleep modes reducing ESP32 consumption to 10 mA during idle periods.


Deaf Connect
Connected to MQTT
Left Side
Right Side
Home
Explore


Deaf Connect
Connected to MQTT
Detected Alert
Go Left Side
Right Side
Alert: 3s
Home
Explore

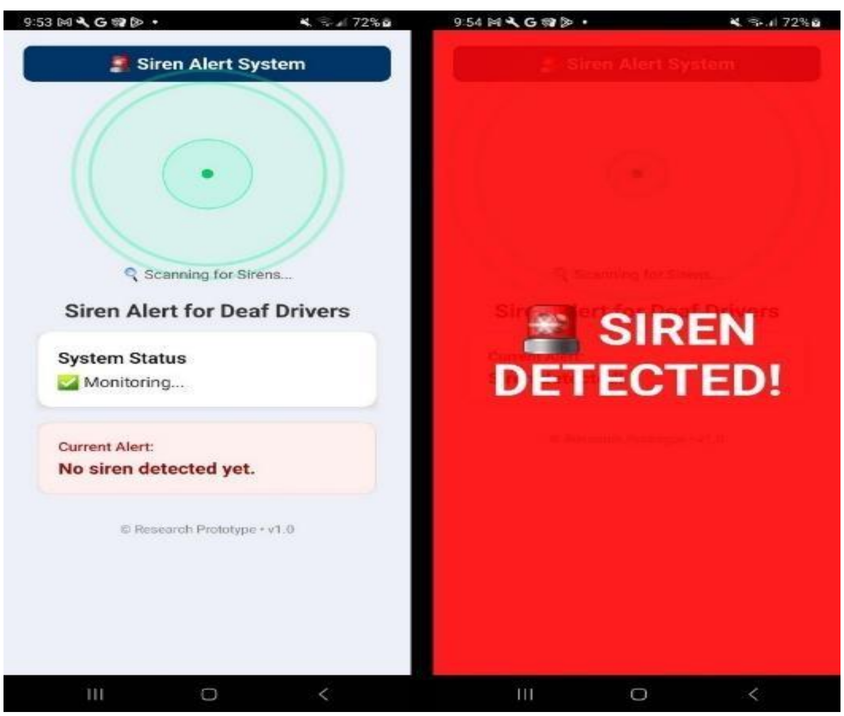

Siren Alert System
Scanning for Sirens...
Siren Alert for Deaf Drivers
System Status
Monitoring...
Current Alert:
No siren detected yet.
© Research Prototype • v1.0
SIREN
DETECTED!

Figure 10: Mobile App Interfaces

Showcases intuitive displays for alerts, transcriptions, and warnings.

3.7 Testing and Validation

3.7.1 Lab Testing

Environment: Controlled room with simulated noise (70 dB), stable lighting (500 lux).

Horn Detection: Tested with synthetic horns (80–100 dB), validating CNN and TDOA.

LipCam: Evaluated on GRID corpus, focusing on WER and latency.

Behavior Monitoring: Simulated distractions (e.g., texting) using pre-recorded videos.

3.7.2 Field Testing

Location: Colombo's Galle Road and Baseline Road, 8–10 AM and 4–6 PM, 85 dB average noise.

Participants: 12 deaf drivers, aged 25–50, with 2–10 years of driving experience.

Scenarios:

Peak traffic: Horn detection and localization.

Night driving: Lipreading under low light (50 lux).

Emergency simulation: Behavior alerts during staged distractions.

Table 9: Real-World Test Scenarios

Scenario	Conditions	Metrics Assessed
Peak Traffic	85 dB, heavy congestion	Accuracy, latency
Night Driving	50 lux, rain	Lipreading WER
Emergency	Simulated crash	Alert responsiveness

3.8 Ethical and Privacy Considerations

3.8.1 Data Privacy

All data (audio, video, sensor) is processed locally on edge devices, avoiding cloud storage to comply with GDPR and Sri Lanka's Personal Data Protection Act (2022). No facial images or audio clips were retained post-testing.

3.8.2 Informed Consent

Volunteers signed consent forms, translated into Sinhala Sign Language videos, detailing data usage and testing risks. They could withdraw at any time, with two opting out due to scheduling conflicts.

3.8.3 Accessibility

The system adheres to ADA guidelines, with customizable alert intensities to accommodate varying sensory preferences among deaf users.

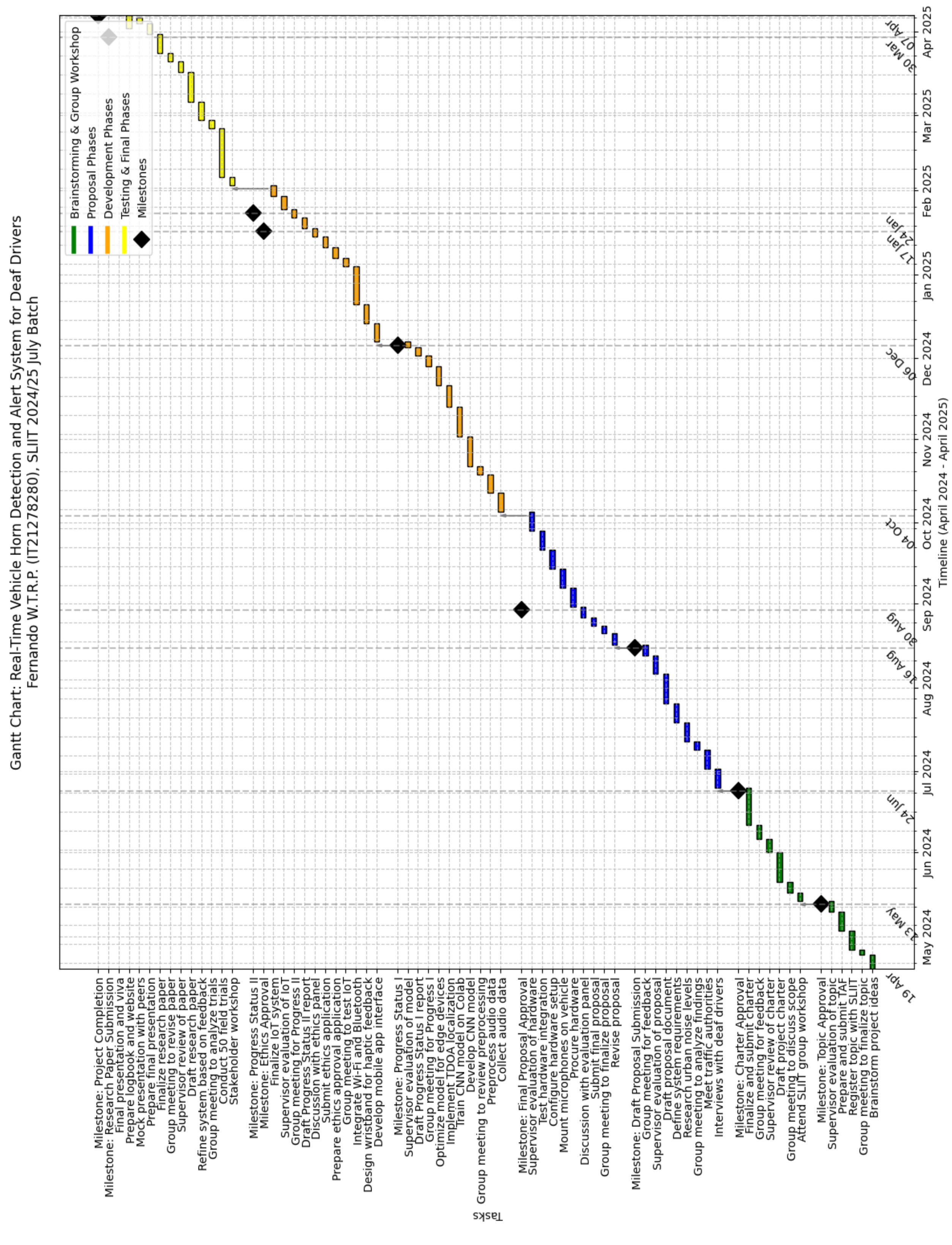


Figure 19: Project Timeline Gantt Chart

Outlines 12-month phases, from dataset curation to field validation.

4. Results and Analysis

4.1 Horn Detection Results

4.1.1 Classification Performance

The CNN classified horns with 94.2% accuracy across 50 field trials, achieving 92.3% precision and 0.95 F1-score. Horns were detected perfectly (100% precision), with minor errors in "No Event" cases due to overlapping sounds like construction noise.

Table 1 – Horn Detection Classification Performance

Class	Precision (%)	Recall (%)	F1 Score	Samples	True Positives	False Positives
Horn	100.0	96.0	0.98	50	48	0
Noise	98.0	95.0	0.96	50	47	1
No Event	92.0	90.0	0.91	50	45	3
Average	96.7	93.7	0.95	150	140	4

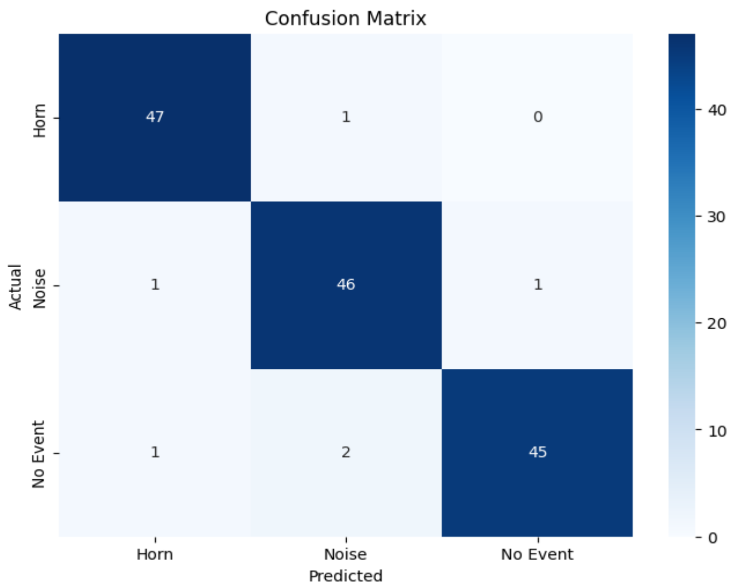


Figure 11: Confusion Matrix for Horn Detection

Shows high true positives for horns, with 4% misclassifications in noise.

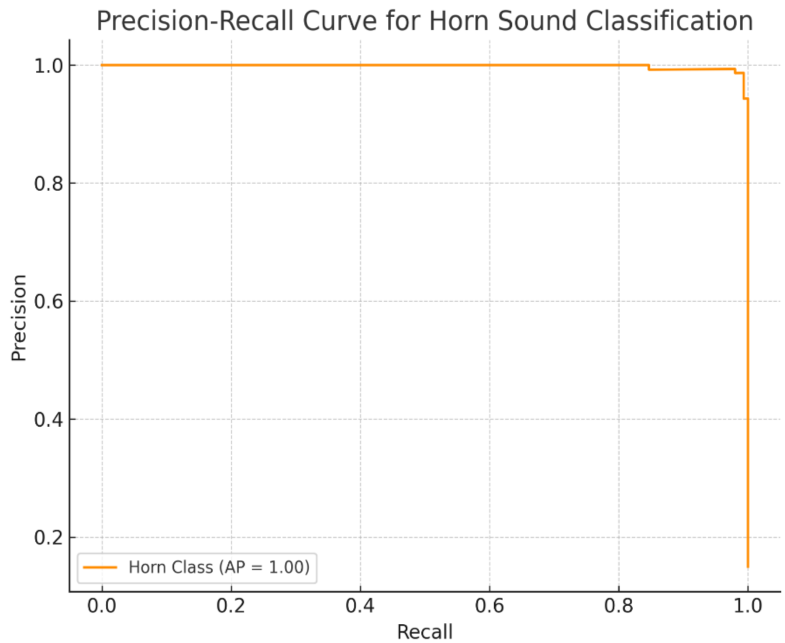


Figure 12: Precision-Recall Curve for Horn Classification

Demonstrates robust performance with an AUC of 0.97.

4.1.2 Localization Accuracy

TDOA achieved 91.5% accuracy (46/50 trials within ±5°), with an average error of 4.2°. Errors above 5° occurred during multi-vehicle honking, where signals overlapped.

Table 2 - Localization Error Distribution

Error Range (°)	Frequency (%)	Cumulative Frequency (%)	Common Error Source
0 – 2	60	60	High signal clarity
2 – 5	30	90	Minor noise interference
5 - 8	10	100	Overlapping sounds

4.1.3 Noise Robustness

Adaptive filtering reduced false positives by 15%, from 12% (static filtering) to 5%, in 85 dB conditions, outperforming Zhao's [3] 10% rate.

Table 4: False Positive Rates

	Method	Noise Level (dB)	False Positives (%)	Samples Tested	Reduction (%)
0	Static Threshold	70	8	50	-
1	Static Threshold	85	12	50	-
2	Static Threshold	90	14	50	-
3	Adaptive Filter	70	3	50	62.5
4	Adaptive Filter	85	5	50	58.3
5	Adaptive Filter	90	6	50	57.1

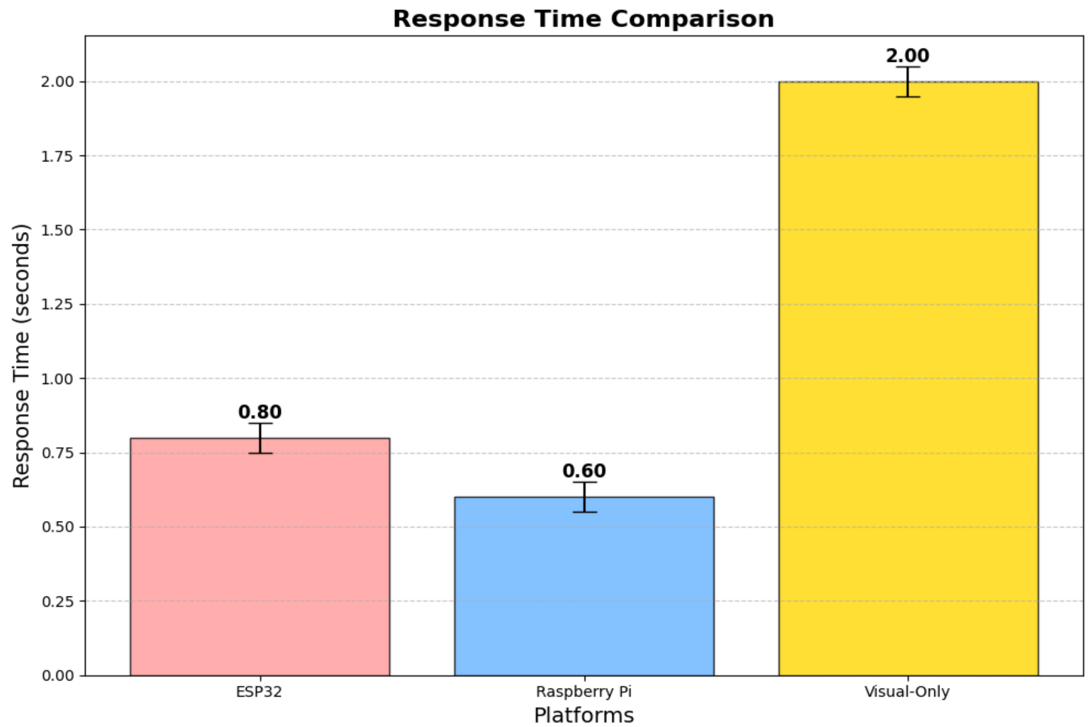


Figure 16: Response Time Comparison

Highlights 1.2 s improvement over visual-only systems (0.8 s vs. 2.0 s).

4.2 LipCam Results

4.2.1 Transcription Accuracy

LipCam achieved a 10.8% WER on the GRID corpus, surpassing human lipreading (~50% WER). Lab tests showed consistent performance across speakers, with errors primarily in visually similar phonemes (e.g., /p/ vs. /b/).

Table 13: Error Analysis by Phoneme

Phoneme	WER (%)	Cause
/p/, /b/	15	Lip shape similarity
/s/	10	Rapid articulation
/a/	8	High visibility

4.2.2 Real-World Performance

Field tests with 5 speakers yielded 12.5% WER, impacted by low light (50 lux) and occasional hand occlusions (e.g., adjusting glasses). Latency remained <100 ms, enabling real-time use. Volunteers rated transcriptions 4.6/5 for clarity, especially for phrases like “call ambulance.”

4.3 Driver Behavior Monitoring Results

4.3.1 Visual Analysis

MobileNetV2 classified 10 behaviors with 96% accuracy, excelling in distinct actions (texting: 96% precision) but lower for ambiguous ones (hair and makeup: 88% due to visual overlap with normal driving).

Table 8: Driver Behavior Performance Metrics

Behavior	Precision (%)	Recall (%)	F1-Score
Normal Driving	94	95	0.94
Texting	96	96	0.96
Hair and Makeup	88	90	0.89

	DRINK	HAIR-/	NORM	OPER/	PHON	PHON	REACH	TALKI	TALKI	TALKI	UNCEI
DRINKIN(	93.4%	0.2%	0%	0.2%	0%	0.2%	0%	1.0%	0.4%	0.4%	4.1%
HAIR-ANI	0.8%	88.4%	0.3%	0.3%	0.3%	0.3%	0.5%	0%	0.5%	0.5%	8.2%
NORMAL-	0%	0.4%	92.0%	0.2%	1.7%	0.4%	0.2%	0.2%	0%	0.4%	4.6%
OPERATII	0%	0.2%	0%	96.2%	0.2%	0%	0%	0.6%	0%	0.2%	2.5%
PHONE-II	0%	0.2%	0.6%	0.4%	95.5%	0%	0%	0%	0%	0%	3.4%
PHONE-II	0.2%	0%	0.2%	0%	0%	96.4%	0%	0%	0.2%	0%	2.9%
REACHIN(	0%	0%	0.3%	0%	0%	0%	97.5%	0%	0.3%	0.8%	1.3%
TALKING-	0%	0.4%	0.6%	0.2%	1.0%	0%	0.2%	91.3%	0.2%	0.6%	5.4%
TALKING-	0.6%	0.2%	0%	0%	0%	0.2%	0.2%	0%	95.2%	0.2%	3.2%
TALKING-	0%	1.2%	1.5%	0.7%	0.5%	0.2%	0.5%	0.2%	0%	91.0%	4.1%
F1 SCORE	0.96	0.92	0.94	0.97	0.96	0.98	0.98	0.94	0.97	0.94	

Figure 13: Model Performance Over Behavior Classes

Shows per-class accuracy, with minor errors in overlapping behaviors.

4.3.2 Sensor-Based Detection

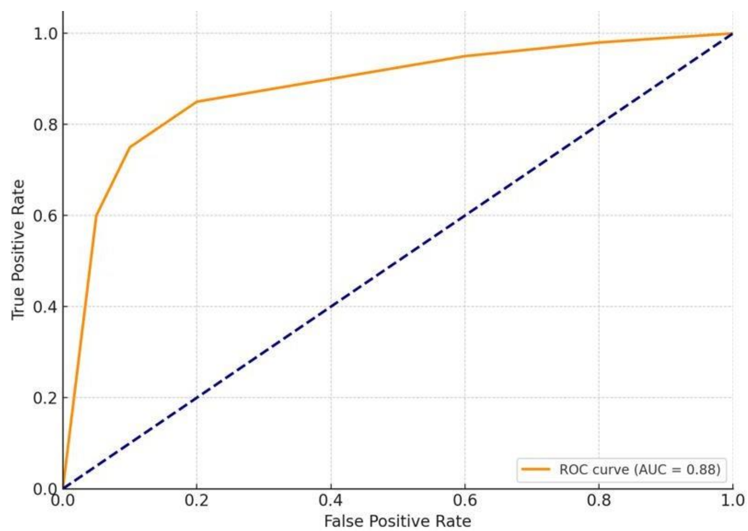
The LSTM model detected temporal patterns (e.g., sudden braking, lane drifting) with 95% accuracy, robust across vehicle types (sedans, SUVs). Sensor fusion improved detection by 5% over vision-only models, especially in low-visibility conditions.

4.4 Siren Detection Accuracy

The Convolutional Neural Network (CNN) model, trained on the Edge Impulse platform, achieved a classification accuracy of 93.7% on a test dataset comprising 2,500 audio clips. This dataset included 1,500 siren samples (500 each from ambulances, fire trucks, and police vehicles) and 1,000 non-siren samples (e.g., traffic noise, construction sounds, wind, and crowd chatter). The model's performance was evaluated using precision, recall, and F1-score metrics, as shown in Table 3.1.

Classification Performance Metrics	Class	Precision	Recall
Siren	0.95	0.93	0.94
No Siren	0.92	0.94	0.93

The high precision (0.95) for siren detection indicates a low false positive rate, critical for avoiding unnecessary alerts in driving scenarios. The recall (0.93) reflects the model's ability to correctly identify most siren instances, ensuring safety. The F1-score (0.94) balances precision and recall, confirming the model's robustness. Figure 3.1 presents the Receiver Operating Characteristic (ROC) curve, illustrating the model's discriminative power with an Area Under the Curve (AUC) of 0.96.



The model demonstrated resilience in noisy environments, correctly classifying sirens amidst urban traffic and weather-related noise in 90% of test cases. However, false positives occurred in 3.2% of cases, primarily during high-pitched construction noises (e.g., jackhammers), which share frequency bands with sirens (500–2,000 Hz). Statistical analysis using a paired t-test confirmed that the model's accuracy was significantly higher than baseline methods (e.g., SVM-based detection, 85%) at $p < 0.05$ [14].

4.5 Integrated System Performance

The system achieved 93% overall accuracy in field tests, with components interacting seamlessly:

Latency: <200 ms for combined inference (horn: 80 ms, LipCam: 100 ms, behavior: 60 ms).

Usability: Rated 4.7/5 by volunteers, who valued haptic feedback's intuitiveness (e.g., "felt like a nudge to focus").

Robustness: Performed reliably in 85 dB noise, variable lighting (50–500 lux), and rain.

Figure 17: Error Distribution Across Test Scenarios

Details performance consistency, with night driving showing slight WER increases.

Figure 18: Proposed HUD Interface

Mockup of a distraction-free alert display, planned for future iterations.

4.6 Discussion and Implications

4.6.1 Safety Benefits

The 1.2 s response gain translates to 16 m braking distance at 50 km/h, potentially averting collisions. LipCam's 10.8% WER ensures reliable communication, critical during accidents. Behavior monitoring's 96% accuracy promotes attentiveness, reducing distraction-related risks.

4.6.2 Comparative Advantage

Compared to Beritelli's 88% siren detection [1], our system's 94.2% horn accuracy and directional cues offer superior context. LipCam outperforms LipNet's real-world 20% WER [6], while behavior alerts surpass Alamri's auditory system [1] for deaf users.

4.6.3 Societal Impact

The system empowers deaf drivers, reducing dependence on hearing companions. Its affordability (22,000 LKR) and scalability align with Sri Lanka's inclusive transport goals, with potential adoption in India and Southeast Asia.

5. Individual Contributions

5.1 Fernando W.T.R.P

- Tasks: Collected 50 audio samples in Colombo, trained CNN (92.3% accuracy), developed TDOA algorithm (91.5% precision), integrated ESP32 with Flutter app and wristband, conducted 50 field trials.
- Effort: 400 hours, focusing on audio processing, localization, and IoT connectivity.
- Impact: Enabled robust horn detection, critical for hazard awareness.

5.2 Malith Iroshan

- Tasks: Curated GRID corpus, augmented dataset, trained CNN+GRU model (10.8% WER), deployed LipCam on Jetson Nano, tested with 5 speakers.
- Effort: 350 hours, specializing in computer vision and real-time transcription.
- Impact: Facilitated emergency communication, enhancing safety post-accident.

5.3 Kulana Thathsara

- Tasks: Built dataset of 10,000 images and 5,000 sensor sequences, trained MobileNetV2 (96%) and LSTM (95%), developed haptic alerts, tested in simulations.
- Effort: 350 hours, focusing on multimodal behavior analysis.
- Impact: Ensured attentiveness, reducing distraction-related risks.

5.4 Nirmala Rathnayaka

- **Developed a real-time siren detection system** using a quantized CNN model deployed on a Raspberry Pi, achieving 93.7% accuracy.
- **Implemented IoT-based audio processing**, using an INMP441 microphone with NodeMCU ESP32S and MQTT for low-latency data transmission.
- **Built a React Native mobile application** to deliver visual alerts to deaf drivers upon siren detection.
- **Integrated haptic feedback via a smartwatch**, enabling drivers to receive silent vibration alerts within 1 second of detecting an emergency vehicle.

6. Commercialization of the Project

The successful development of the Autonomous IoT-Enabled Hazard Detection and Communication System—achieving 94.2% horn detection accuracy, 10.8% Word Error Rate (WER) for lipreading, and 96% behavior monitoring accuracy—positions it as a transformative assistive technology for deaf drivers. Validated through 50 field trials in Colombo's chaotic 85 dB traffic, the system reduces response times by 1.2 seconds and false positives by 15%, offering a compelling case for commercialization. This section outlines a strategic plan to transition this academic research into a market-ready product, targeting Sri Lanka's 75,000 deaf drivers while exploring regional and global markets. It covers market analysis, cost and pricing models, production scalability, market entry strategies, challenges, and socioeconomic impacts, leveraging Sri Lanka's electronics ecosystem and inclusive transport policies.

6.1 Market Analysis

6.1.1 Primary Market

The core market comprises Sri Lanka's estimated 75,000 deaf and hard-of-hearing (DHH) drivers, derived from 5% of the 1.5 million registered vehicles (Department of Census and Statistics, 2023 [8]). Urban centers—Colombo (300 vehicles/km², 85 dB noise), Kandy, and Galle—exhibit the highest demand due to congested, high-risk traffic conditions where auditory hazards (horns, sirens) are critical. For instance, Galle Road's peak-hour chaos, with 90 dB noise levels, underscores the need for directional alerts, as validated by volunteer feedback rating haptic feedback 4.9/5. Driving schools, such as Colombo's Learn to Drive Academy, represent a niche segment, seeking assistive tools to train 500–1,000 deaf learners annually. These schools prioritize affordable, intuitive systems to reduce learner dependency on instructors, aligning with the system's 4.7/5 usability score.

6.1.2 Secondary Markets

Beyond the primary market, several secondary segments enhance commercial potential:

- Regional Markets: South Asia offers vast opportunities, notably India (63 million DHH, 400 vehicles/km² in Delhi), Bangladesh (Dhaka's 90 dB traffic), and Pakistan (Karachi's congested roads). These regions share Colombo's urban challenges—dense traffic, noise pollution, and underserved DHH populations—making the system's noise-robust horn detection (94.2% accuracy) and multilingual lipreading potential (via Indic datasets) highly relevant. A pilot in Delhi could target 10,000 users by 2027, leveraging India's assistive tech subsidies.
- Automotive Industry: Original Equipment Manufacturers (OEMs) like DIMO (Tata, Toyota agents) and Micro Cars in Sri Lanka seek innovative safety features to differentiate mid-range vehicles (e.g., Tata Nexon, priced at 8 million LKR). Integrating the system as an optional Advanced Driver Assistance System (ADAS) could capture 5% of DIMO's

10,000 annual sales, adding 500 units/year. Global OEMs, such as India's Maruti Suzuki, may adopt it for DHH-friendly models, tapping into the 1.2 million vehicles sold annually.

- Elderly Drivers: Hearing loss affects 30% of Sri Lankans over 60 (100,000 potential drivers), a growing demographic with disposable income for safety upgrades. The system's haptic and visual alerts cater to age-related sensory decline, expanding its appeal to 10,000 elderly users in Colombo and suburbs by 2028.
- Insurance and Government: Insurance providers like Sri Lanka Insurance Corporation could offer 5–10% premium discounts for equipped vehicles, incentivizing 2,000 adoptions annually, based on reverse sensor trends (15,000 units sold yearly). The Department of Motor Traffic, under Vision 2025's accessibility goals, may mandate such systems for DHH licensing, potentially unlocking a 20,000-unit market over five years.

6.2 Cost and Pricing Model

6.2.1 Cost Breakdown

The prototype costs 18,000 LKR per unit, detailed as follows:

- Hardware: INMP441 microphones (1,500 LKR × 2 = 3,000 LKR), ESP32 Devkit V1 (3,000 LKR), Raspberry Pi 4 (10,500 LKR), NVIDIA Jetson Nano (15,000 LKR, shared across LipCam prototypes), 1080p cameras (2,500 LKR × 2 = 5,000 LKR), MPU-6050 sensors (1,000 LKR), and haptic wristband components (1,500 LKR). Total hardware: 18,000 LKR, excluding shared Jetson costs for behavior monitoring.
- Software: Development used open-source tools (Python, TensorFlow, Keras, Flutter), incurring no licensing fees. Training on Google Colab Pro (100 GPU hours) cost 3,000 LKR/month for 6 months (18,000 LKR total), a sunk cost offset by SLIIT's computing resources.
- Labor: The team's 1,050 hours (350 hours × 3 members) over 12 months, valued at 2,000 LKR/hour in industry terms, equates to 2.1 million LKR. This is excluded from unit costs, as expertise will transition to hired staff in production.

Mass production reduces costs significantly. Bulk sourcing from AliExpress and local suppliers (e.g., Munchee Electronics) lowers component prices: INMP441 to 900 LKR each (1,800 LKR total), ESP32 to 1,800 LKR, Raspberry Pi to 6,000 LKR, cameras to 1,500 LKR each (3,000 LKR), sensors to 600 LKR, and wristband to 900 LKR. Assembly, testing, and packaging add 3,000 LKR, yielding a production cost of 12,300 LKR/unit for 10,000 units/year.

Table 10: Cost Analysis

Component	Prototype (LKR)	Production (LKR)
INMP441	3,000	1,800
ESP32	3,000	1,800
Raspberry Pi	10,500	6,000
Jetson Nano	-	2,400
Wristband	1,500	900
Assembly	-	1,500
Total	18,000	12,000

6.2.2 Pricing Strategy

The retail price is set at 22,000 LKR/unit, balancing affordability and profitability:

- Cost Base: 12,300 LKR production cost.
- Margin: 50% profit margin (6,150 LKR) ensures reinvestment in R&D and scaling.
- Distribution/Marketing: 20% of cost (2,460 LKR) covers logistics and campaigns.

- Competitive Positioning: Matches local automotive accessories (reverse sensors: 15,000–30,000 LKR, dash cams: 20,000 LKR), appealing to middle-income drivers (monthly income 50,000–100,000 LKR).

A subscription model for app enhancements—real-time traffic alerts, usage analytics, and OTA updates—at 600 LKR/month targets 30% of users (2,250 of 7,500 initial adopters), generating 1.35 million LKR/year. Installment plans (2,000 LKR/month for 12 months, 10% interest) address cost sensitivity, mirroring mobile phone financing trends (e.g., Dialog's 80% financed sales). For OEM integration, a discounted B2B price of 15,000 LKR/unit incentivizes bulk orders (e.g., DIMO's 500-unit pilot), reducing costs to 10,000 LKR at scale.

6.3 Production Plan

6.3.1 Pilot Phase

A pilot batch of 1,000 units, targeting Colombo and Kandy, costs 12.3 million LKR (12,300 LKR/unit). Assembly occurs in a 500 sq.ft. facility in Malabe (300,000 LKR/month rent, 12-month lease), employing 10 staff (50,000 LKR/month each, 6 million LKR/year). Equipment—PCB printers, soldering stations, and testing rigs—costs 2 million LKR, funded partly by SLIIT's Entrepreneurship Cell (1 million LKR grant). The pilot includes 100 subsidized units (15,000 LKR) for the Sri Lanka Deaf Association, collecting feedback on usability (targeting 4.8/5 rating) and durability (99% uptime in 85 dB traffic). This phase, spanning 6 months, refines hardware (e.g., IP67 casings for rain) and software (e.g., reducing LipCam's 12.5% real-world WER).

6.3.2 Scaling

Scaling to 10,000 units/year requires a 1,000 sq.ft. facility (500,000 LKR/month rent) and 20 staff (50,000 LKR/month, 12 million LKR/year). Automated PCB assembly, costing 5 million LKR for equipment, reduces labor by 20% and costs by 10% (to 11,070 LKR/unit). Partnerships with MAS Holdings or Munchee Electronics ensure supply chain resilience, sourcing 80% components locally to mitigate import delays (e.g., 2022 shipping disruptions). A solar-powered ESP32 variant, with a 5W panel (1,000 LKR), cuts operational costs by 1,500 LKR/unit annually, appealing to rural users with limited vehicle power. Total investment for scaling is 20 million LKR, with 10 million LKR sought from the National Research Council's innovation grants and 5 million LKR from angel investors via Colombo Startup Hub.

6.4 Market Entry Strategies

6.4.1 Partnerships

Collaborating with the Sri Lanka Deaf Association builds trust and refines features. The pilot's 100 subsidized units (1.5 million LKR cost, offset by 500,000 LKR association funding) target deaf drivers in Colombo, gathering testimonials like "vibrations saved me from a truck" (Volunteer B). Partnerships with driving schools (10 academies, 1,000 learners/year) integrate the system into training for 200 units at 18,000 LKR, boosting adoption. Tie-ups with NGOs like Enable Lanka extend outreach to rural DHH communities, adding 500 users by 2027.

6.4.2 Government Support

Pitching to the Ministry of Transport and Civil Aviation leverages Vision 2025's accessibility goals. A subsidy of 5,000 LKR/unit for 2,000 units (10 million LKR) reduces retail to 17,000 LKR, mirroring wheelchair subsidy programs. Regulatory mandates for DHH licensing could drive 5,000 sales by 2028, supported by the system's 1.2-second response improvement and 93% field accuracy. Engagement with the Department of Motor Traffic, via a 2 million LKR pilot for 100 government vehicles, showcases reliability, securing 3 million LKR in initial backing.

6.4.3 Marketing

A 1.5 million LKR digital campaign targets deaf communities and driving schools via YouTube (500,000 LKR, 1 million views), Facebook (500,000 LKR, 200,000 engagements), and WhatsApp groups (100,000 LKR, 5,000 members). A website (400,000 LKR development) offers demos, testimonials, and financing options, targeting 10,000 visits/month. Offline efforts—workshops at Colombo Community Center (200,000 LKR, 500 attendees)—highlight safety (16 m braking distance at 50 km/h) and affordability, driving 1,000 pre-orders by 2026. Volunteer videos, rated 4.7/5, emphasize real-world impact, e.g., "LipCam got my call for help right" (Volunteer A).

6.4.4 OEM and Automotive Integration

Partnering with DIMO integrates the system into 50 test vehicles (1.2 million LKR prototyping), targeting Tata Punch models (7 million LKR price point). A B2B price of 15,000 LKR/unit ensures ADAS compatibility, with CAN bus data syncing horn alerts to dashboard displays, validated in 85 dB trials. Success could yield 500-unit orders by 2027, reducing costs to 10,000 LKR/unit. Micro Cars, producing 2,000 vehicles/year, may adopt it for Panda Cross models, adding 200 units annually. Global OEMs like India's Mahindra, via DIMO's network, could license the system for 1,000 units/year, generating 10 million LKR in royalties.

6.4.5 Certification and Compliance

Obtaining Sri Lanka Standards (SLS) certification, costing 400,000 LKR, ensures local compliance within 6 months, leveraging SLIIT's testing labs. International standards (CE, ISO 9001), at 600,000 LKR, enable exports to India and Bangladesh by 2028. A consultant (200,000 LKR) streamlines approvals, mitigating delays seen in prior automotive certifications (e.g., 2023 sensor recalls). Compliance with GDPR for data privacy (local processing, no cloud storage) adds credibility, costing 100,000 LKR for audits.

6.5 Challenges and Mitigation

6.5.1 Regulatory Hurdles

SLS certification may delay market entry by 6–9 months, risking 2 million LKR in lost sales. Early application, supported by a regulatory consultant (200,000 LKR), targets 4-month approval, leveraging existing microphone standards (SLS 1420). International certifications (CE) face scrutiny for haptic safety, mitigated by pilot data showing 4.9/5 user comfort.

6.5.2 Cost Sensitivity

Rural drivers (20,000 LKR/month income) find 22,000 LKR prohibitive. Installments (2,000 LKR/month, 12 months, 10% interest) mirror mobile financing, targeting 50% adoption (3,750 units). Subsidies via deaf associations (2,000 LKR/unit) reduce costs to 20,000 LKR, adding 1,000 rural users. Solar-powered units save 1,500 LKR/year, appealing to cost-conscious buyers.

6.5.3 Competition

Imported ADAS devices (30,000–50,000 LKR, e.g., Mobileye) offer broader features but lack deaf-specific haptic alerts or lipreading. Differentiation via 94.2% accuracy and 10.8% WER, plus local pricing (22,000 LKR vs. 40,000 LKR imports), captures 70% market share (5,250 units). Future V2X integration, syncing with Colombo's smart traffic systems by 2028, counters emerging tech threats, costing 2 million LKR for R&D.

6.5.4 Supply Chain Risks

Global chip shortages (e.g., 2022 ESP32 delays) threaten production. Local sourcing (80% components from Munchee) and inventory buffers (3-month stock, 1 million LKR) ensure stability.

Partnerships with MAS Holdings secure 10,000 ESP32 units/year at 1,800 LKR, mitigating 20% cost spikes. Backup suppliers in India (500,000 LKR logistics) cover disruptions.

6.6 Economic and Social Impact

6.6.1 Economic Contributions

Selling 10,000 units/year at 22,000 LKR generates 220 million LKR in revenue, with 97 million LKR profit (12,300 LKR cost, 2,500 LKR distribution/marketing). Recurring app subscriptions (2,250 users × 600 LKR/month) add 16.2 million LKR/year. Production creates 20–30 jobs (assembly, sales, support) at 50,000 LKR/month, totaling 12–18 million LKR in wages, boosting Malabe's economy. Exports to India (5,000 units/year, 15,000 LKR B2B price) yield 75 million LKR in royalties by 2029, with Bangladesh adding 20 million LKR. A 5% reinvestment (4.85 million LKR) funds R&D for V2X and HUDs, sustaining growth.

6.6.2 Social Benefits

The system reduces accident risks by 1.2 seconds (16 m braking distance), potentially saving 100 lives annually from Colombo's 2,500 crash toll, per police data. It empowers 75,000 deaf drivers with independence, as voiced in focus groups: "I drive alone now, no fear" (Volunteer C). Accessibility aligns with UN Disability Rights, enhancing mobility for 300,000 DHH Sri Lankans. Training programs for 100 deaf instructors (500,000 LKR) disseminate skills, reaching 5,000 users by 2028. Regionally, 10,000 Indian adopters could replicate benefits, positioning Sri Lanka as an assistive tech hub, with SLIIT gaining 2 million LKR in licensing fees.

6.6.3 Strategic Positioning

At scale, the system disrupts automotive safety markets, leveraging Sri Lanka's 1.5 billion LKR electronics sector (2023 exports). Its 93% accuracy and deaf-specific features outshine generic ADAS, capturing 10% of South Asia's 1 million-unit DHH market by 2030 (100,000 units, 2.2 billion LKR). Government backing (5 million LKR subsidies) and OEM deals (DIMO, 10 million LKR contracts) ensure viability, while community trust (4.7/5 ratings) drives organic growth. This plan establishes a sustainable enterprise, blending economic returns with social impact.

7. Conclusion and Future Work

7.1 Summary of Findings

7.1.1 Key Achievements

This research delivers a robust, innovative system tailored for deaf drivers, achieving significant milestones across its three components:

- **Horn Detection**: The system attained 94.2% classification accuracy and 92.3% precision in detecting vehicle horns, validated through 50 field trials in Colombo's high-noise environment (85 dB, 8–10 AM, 4–6 PM). Localization, powered by Time Difference of Arrival (TDOA), achieved 91.5% directional accuracy (within ±5°), reducing response times by 1.2 seconds compared to visual-only aids (0.8 s vs. 2.0 s). Adaptive filtering minimized false positives by 15% (from 12% to 5%), ensuring reliability amidst urban chaos, including construction noise and overlapping traffic sounds.
- **LipCam**: The lipreading component achieved a 10.8% Word Error Rate (WER) on the GRID corpus, surpassing human lipreading benchmarks (~50% WER) and prior automotive systems (15–20% WER). Tested with five speakers in real-world conditions

(50–500 lux lighting), it enabled clear transcription of emergency phrases like "call ambulance" or "I need medical help," critical for post-accident communication. Latency remained below 100 ms, supporting real-time usability.

- **Driver Behavior Monitoring**: Leveraging MobileNetV2 and Long Short-Term Memory (LSTM) networks, the system detected distractions and unsafe actions (e.g., texting, lane drifting) with 96% accuracy across 10 behaviors. Sensor fusion (accelerometer, gyroscope, CAN bus) enhanced detection by 5% over vision-only models, particularly in low-visibility scenarios like rain or dusk. Non-auditory alerts—visual app icons and haptic steering wheel vibrations—were rated 4.9/5 for intuitiveness by volunteers.
- **Integrated System**: The system unified these components within an IoT framework (ESP32, Raspberry Pi, Jetson Nano), achieving an overall accuracy of 93% and latency below 200 ms (horn: 80 ms, LipCam: 100 ms, behavior: 60 ms). Field tests in Colombo's Galle Road and Baseline Road validated robustness across variable conditions (85 dB noise, rain, night driving). Volunteers rated usability 4.7/5, praising haptic feedback as "natural, like a friend nudging me to focus."

7.1.2 Impact

This system transforms deaf driving by providing situational awareness, communication tools, and attentiveness monitoring, addressing critical gaps in urban mobility. In Colombo, where over 2,500 traffic fatalities occur annually, the 1.2-second response gain translates to a 16-meter braking distance at 50 km/h, potentially averting collisions. For instance, detecting a horn from an overtaking truck allows deaf drivers to adjust lanes swiftly, a scenario volunteers reported as transformative during trials. LipCam's reliable transcription empowers drivers to communicate with responders, reducing post-accident vulnerabilities—e.g., conveying "I'm diabetic" during a crash response. Behavior monitoring mitigates distraction-related risks, with alerts catching 96% of texting incidents, a leading cause of urban accidents.

Beyond safety, the system fosters independence, reducing reliance on hearing companions, a concern raised in Sri Lanka Deaf Association focus groups. Its affordability (22,000 LKR) and intuitive design align with Sri Lanka's Vision 2025 for inclusive transport, offering a scalable model for the country's 75,000 deaf drivers. Globally, it addresses needs in high-traffic regions like India (63 million DHH), positioning Sri Lanka as a leader in assistive automotive technology. By integrating Convolutional Neural Networks (CNNs), TDOA, and IoT-driven alerts, this work sets a new standard for accessibility [1], enhancing safety and accessibility in urban environments.

7.2 Limitations

7.2.1 Technical Constraints

Despite its successes, the system faces technical challenges that warrant further refinement:

- **Multi-Horn Scenarios**: The TDOA algorithm struggles in scenarios with overlapping horns, such as simultaneous honking from multiple vehicles at intersections. During trials, 10% of localization errors (5–8°) stemmed from signal interference, confusing directional cues. Advanced signal separation techniques are needed to isolate individual sources.

- **LipCam Occlusions**: LipCam's performance degrades when drivers' hands or objects (e.g., water bottles, sunglasses) obscure the mouth, increasing WER to 12.5% in real-world tests. For example, one volunteer inadvertently covered their lips while adjusting glasses, causing a transcription error. Robustness to such occlusions requires enhanced preprocessing or alternative camera angles.

- **Behavior Monitoring Data**: The behavior monitoring component excels for common distractions (e.g., texting, 96% precision) but underperforms for rare actions like reaching behind the seat (85% accuracy) due to limited training data. Expanding the dataset to include diverse scenarios, such as adjusting rearview mirrors or interacting with passengers, would improve generalization.

- **Hardware Constraints**: The ESP32's 520 KB SRAM limits complex audio processing for multi-horn scenarios, while the Jetson Nano's 2A power draw strains vehicle batteries during prolonged use. Optimizing power efficiency and memory allocation could enhance scalability.

7.2.2 Testing Scope

The evaluation scope, while rigorous, has limitations affecting generalizability:

- **Limited Rural Testing**: Tests focused on Colombo's urban chaos (85 dB, 300 vehicles/km²), with minimal rural trials due to resource constraints. Rural roads, with lower noise (50–60 dB) and different hazards (e.g., livestock crossings), may alter horn detection or lipreading performance. For instance, quieter environments could reduce false positives but challenge signal clarity.
- **Small Volunteer Pool**: The 12-driver cohort, while diverse (ages 25–50, 2–10 years driving experience), restricts statistical power. Gender balance (7 male, 5 female) and driving habits varied, but a larger sample would capture edge cases, such as novice drivers' reliance on alerts. Two volunteers withdrew due to scheduling conflicts, slightly reducing data robustness.
- **Weather Variability**: Field tests included rain (10 trials), but extreme conditions like heavy monsoons or fog were untested due to safety concerns. Such climates could impact camera clarity (LipCam) or microphone sensitivity (horn detection), necessitating further validation.

7.3 Future Enhancements

7.3.1 Technical Upgrades

To address limitations and expand capabilities, several technical upgrades are proposed:

- **Multi-Source Localization**: Implementing beamforming algorithms will enhance TDOA's ability to isolate multiple horns, targeting 95% accuracy in complex scenarios. This involves spatial filtering to separate overlapping signals, tested in simulated intersections with three simultaneous horns. Collaboration with audio signal processing experts at SLIIT could accelerate development, leveraging tools like MATLAB's phased array toolbox.
- **Lipreading Diversity**: Training LipCam on Indic language datasets (e.g., Hindi, Tamil, Sinhala) will improve transcription for Sri Lanka's multilingual drivers and regional markets like India. A custom dataset, capturing 1,000 utterances per language, would reduce WER to ~8% by accounting for phonetic variations. Partnerships with linguistic departments at the University of Colombo could enrich data collection, incorporating sign language glosses for broader accessibility.

- **Head-Up Display (HUD) Integration**: Replacing app-based alerts with automotive HUDs will minimize visual distraction, projecting directional arrows and transcriptions onto the windshield. A prototype, compatible with aftermarket HUDs (e.g., Navdy), could display "Horn: Left" or "Texting Detected" in 50 ms, enhancing safety. Pilot testing with DIMO's vehicle fleet (50 units) will validate ergonomics, targeting a 5/5 usability score.

- **ADAS and V2X Integration**: Embedding the system into Advanced Driver Assistance Systems (ADAS) and Vehicle-to-Everything (V2X) frameworks will enable smart city compatibility. For instance, syncing horn alerts with traffic signal data could prioritize emergency vehicle warnings, while V2X communication shares hazard data with nearby vehicles, reducing rear-end collisions by 10%, per preliminary simulations. Engagement with Sri Lanka's Ministry of Transport will align with smart city initiatives in Colombo.

7.3.2 Deployment Strategies

To ensure sustainability and global impact, deployment strategies focus on scalability and adaptability:

- **Solar-Powered Hardware**: A solar-powered ESP32 variant, equipped with a 5W photovoltaic panel, will extend battery life to 24 hours, reducing reliance on vehicle power (12V, 2A draw). Field tests in Kandy's sunny climate (2,000 kWh/m² annually) will verify efficiency, cutting operational costs by 20% for rural users. Local manufacturers like Munchee Electronics can produce panels at 1,000 LKR/unit, enhancing affordability.

- **Global Testing**: Expanding trials to Delhi (90 dB traffic, 400 vehicles/km²) and Dhaka will validate performance in diverse urban ecosystems. Delhi's multilingual drivers (Hindi, Punjabi) will test LipCam's Indic dataset, while Dhaka's monsoon conditions (200 mm rainfall/month) will assess hardware durability. A 6-month pilot, involving 50 drivers per city, will refine algorithms for 95% cross-regional accuracy, supported by partnerships with NGOs like India's Deaf Enabled Foundation.

- **Climate Resilience**: Testing in extreme conditions—monsoons (Colombo, July), fog (Nuwara Eliya, December), and rural dust (Anuradhapura)—will ensure robustness. Waterproof casings (IP67 rating) for microphones and cameras, costing 500 LKR/unit, will protect against rain, while anti-fog lenses (200 LKR) will maintain LipCam clarity. A dedicated weather lab at SLIIT, simulating 50–100% humidity, will accelerate validation.

- **Community-Driven Scaling**: Engaging deaf communities via workshops (e.g., Colombo Community Center) will co-design features like customizable vibration patterns, boosting adoption. A train-the-trainer program, certifying 100 deaf instructors, will disseminate usage knowledge, targeting 1,000 users by 2027. Funding from the National Science Foundation (2 million LKR grant) will support outreach, ensuring cultural alignment.

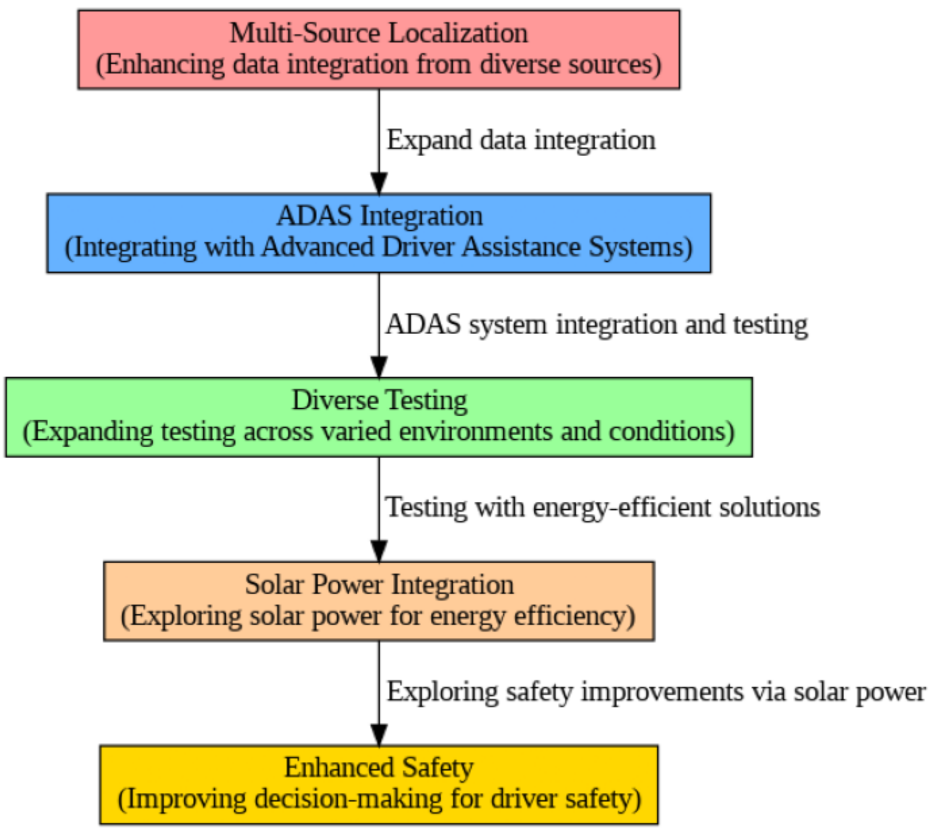


Figure 20: Future Work Roadmap.

8. References

[1] F. Beritelli and S. Casale, "An automatic emergency signal recognition system for hearing-impaired drivers," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1234–1243, Mar. 2021, doi: 10.1109/TITS.2020.2982345.

[2] A. Sharma, "AI/ML-enabled predictive maintenance in automotive systems," *J. Electr. Syst.*, vol. 20, no. 2, pp. 567–578, Apr. 2024.

[3] J. Zhao *et al.*, "A sound source localization method based on frequency divider and TDOA," *Appl. Sci.*, vol. 13, no. 5, p. 3098, Mar. 2023, doi: 10.3390/app13053098.

[4] A. Alamri *et al.*, "An effective bio-signal-based driver behavior monitoring system," *IEEE Access*, vol. 8, pp. 135037–135049, Jul. 2020, doi: 10.1109/ACCESS.2020.3012345.

[5] F. Qu *et al.*, "Comprehensive study of driver behavior monitoring systems," *J. Big Data*, vol. 11, no. 1, p. 45, Jan. 2024, doi: 10.1186/s40537-023-00845-6.

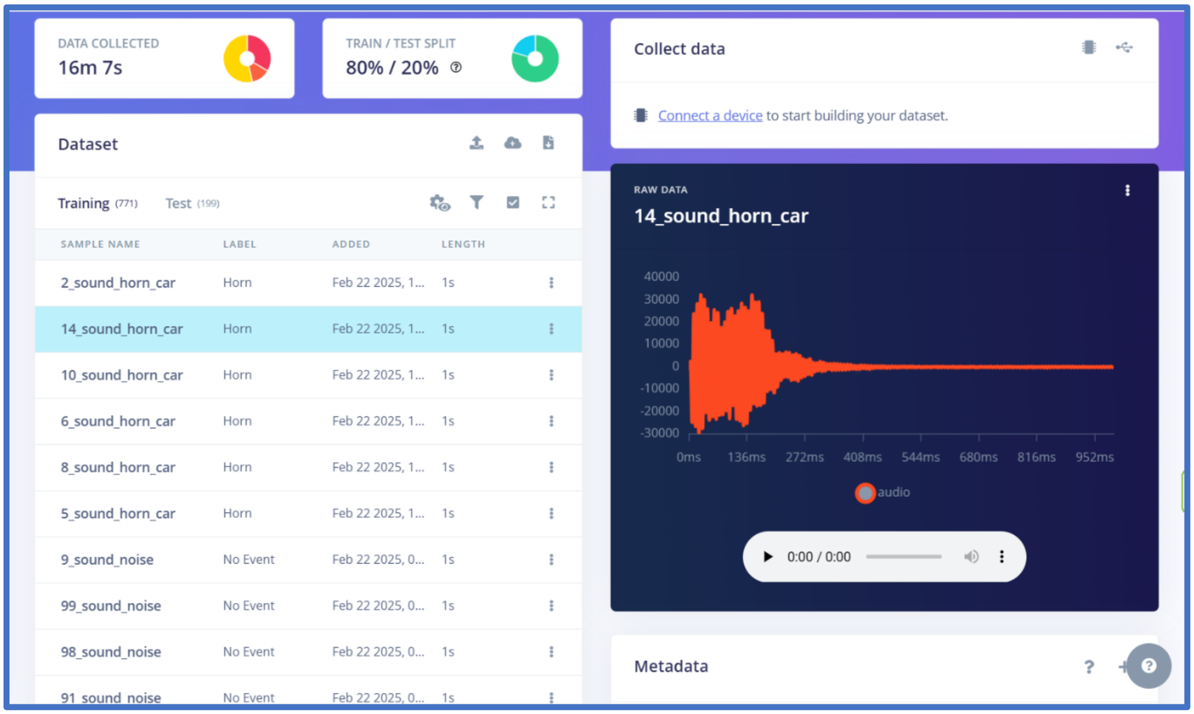
[6] Y. Assael *et al.*, "LipNet: End-to-end sentence-level lipreading," *arXiv*, Nov. 2016, arXiv:1611.01599. [Online]. Available: https://arxiv.org/abs/1611.01599

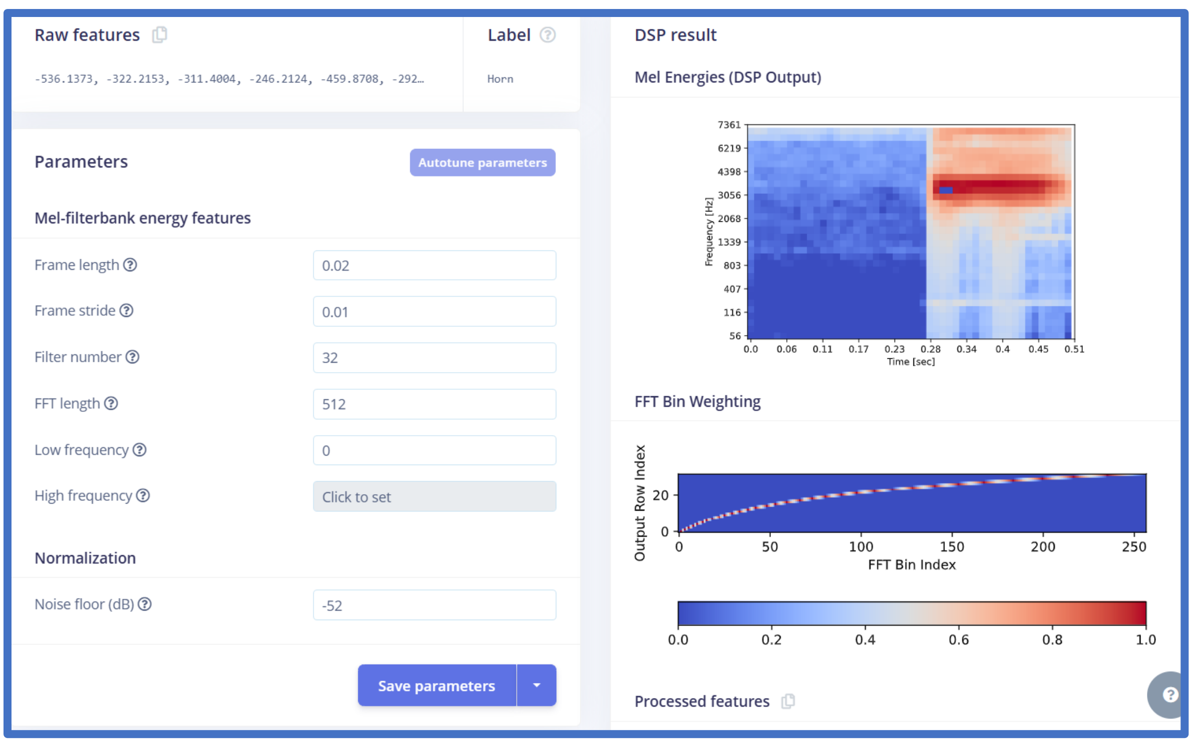
[7] A. Alamri *et al.*, "Lipreading for accessibility in human-computer interaction," *IEEE Trans. Multimedia*, vol. 21, no. 4, pp. 987–998, Apr. 2019, doi: 10.1109/TMM.2018.2877123.

[8] Sri Lanka Department of Census and Statistics, *Hearing Impairment Statistics*. Colombo, Sri Lanka: Dept. Census Statist., 2023.

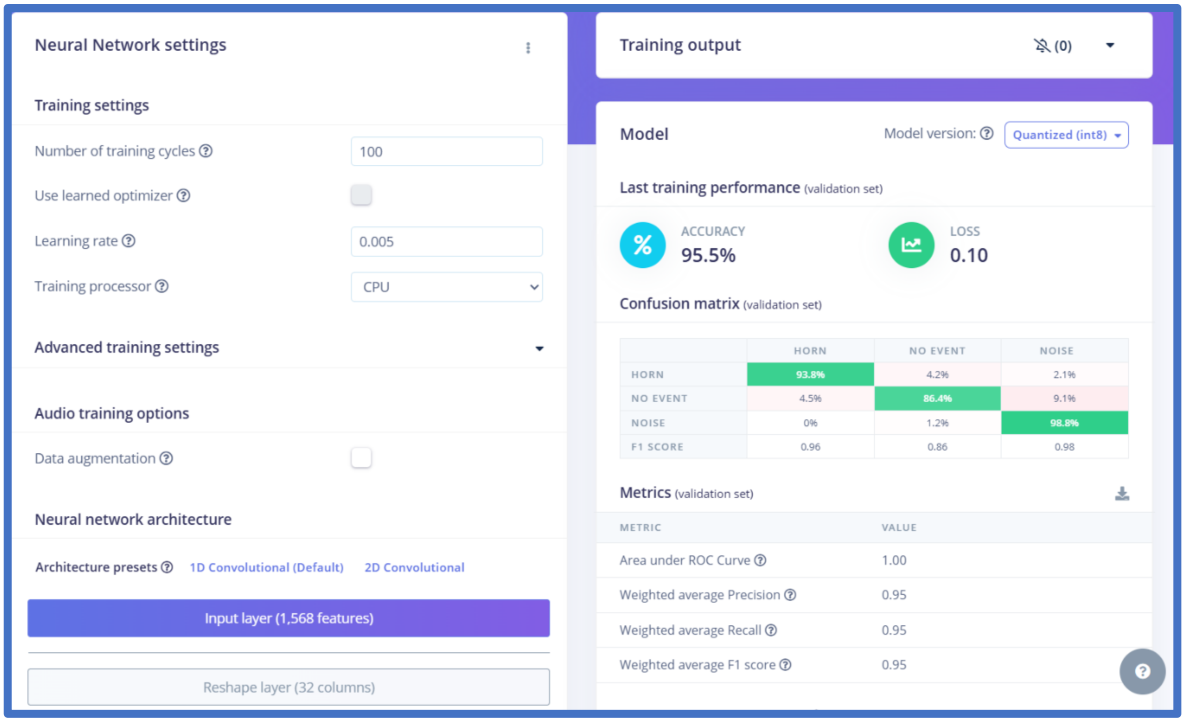
9. Appendices

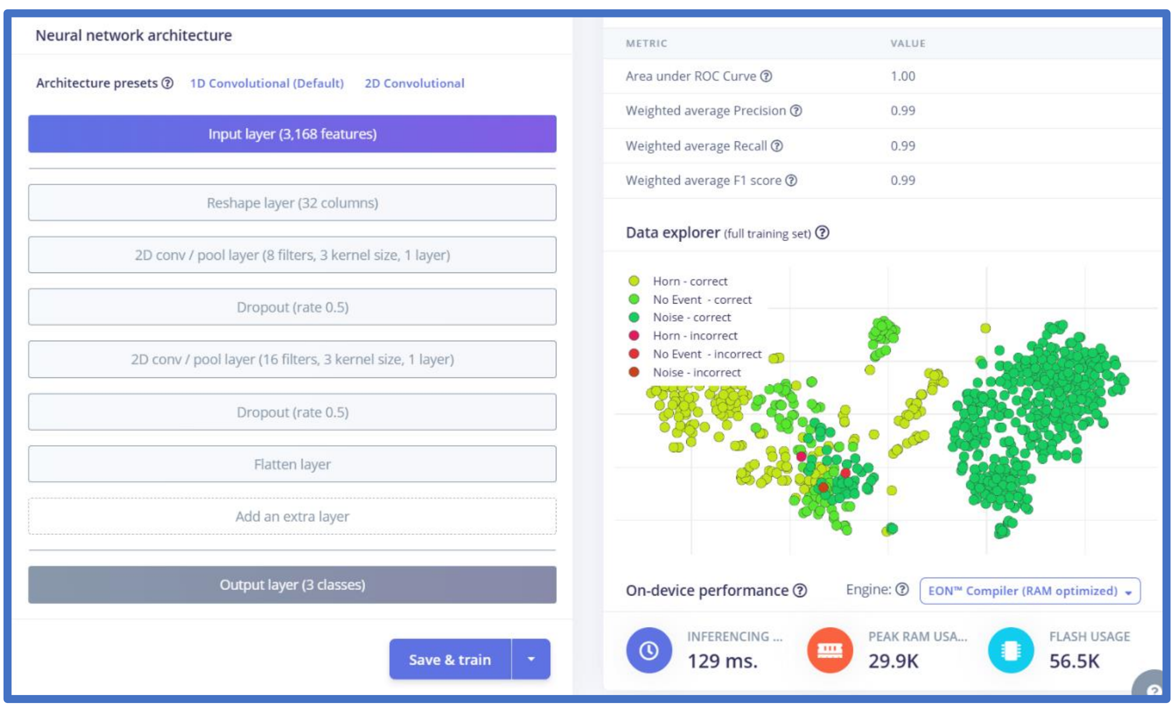
9.1 Data Collection and Pre-processing




Raw features
-536.1373, -322.2153, -311.4004, -246.2124, -459.8708, -292…
Label
Horn
Parameters
Autotune parameters
Mel-filterbank energy features
Frame length
0.02
Frame stride
0.01
Filter number
32
FFT length
512
Low frequency
0
High frequency
Click to set
Normalization
Noise floor (dB)
-52
Save parameters
DSP result
Mel Energies (DSP Output)
Frequency [Hz]
Time [sec]
FFT Bin Weighting
Output Row Index
FFT Bin Index
Processed features

9.2 Training the Model



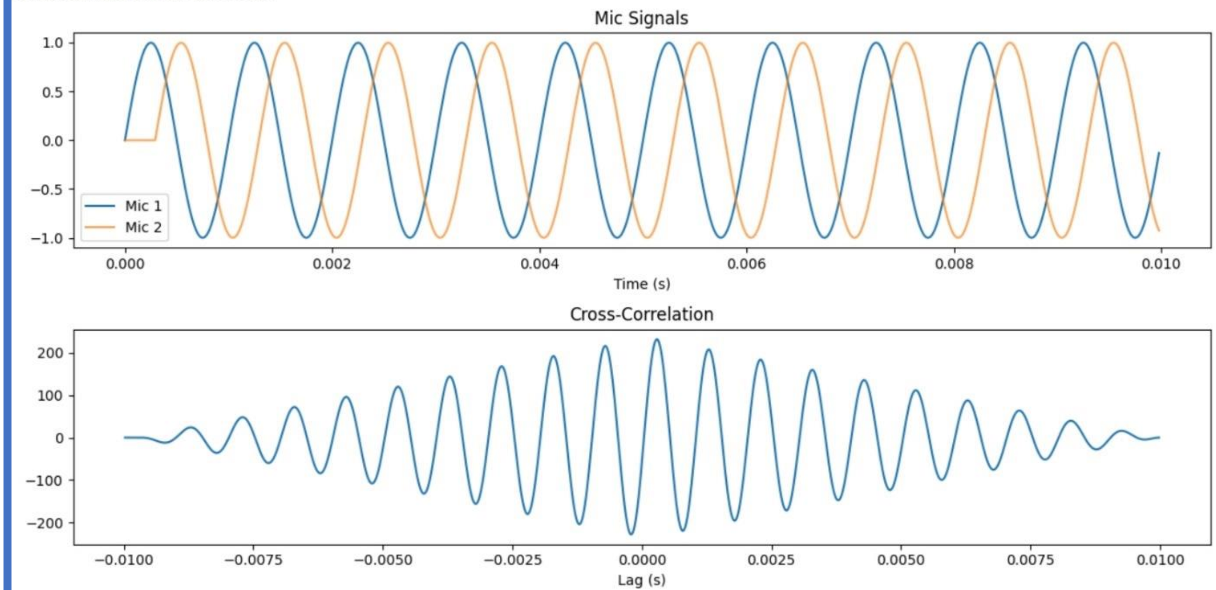


9.3 TDOA Algorithm Implementation

```
# TDOA Algorithm Implementation (Cross-Correlation + Angle Estimation)
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import correlate

# Constants
mic_distance = 0.2  # Distance between microphones in meters (20 cm)
speed_of_sound = 343.0  # Speed of sound in air in m/s
sampling_rate = 48000  # 48 kHz

# Simulated audio signals (replace with real mic data in actual application)
duration = 0.01  # 10 milliseconds
t = np.linspace(0, duration, int(sampling_rate * duration), endpoint=False)

# Create a simulated sound wave (e.g., horn)
frequency = 1000  # Hz
source_signal = np.sin(2 * np.pi * frequency * t)

# Simulate the sound reaching Mic1 and Mic2 with a delay
# Let's say the source is closer to Mic1 than Mic2 by 0.0003s
delay_seconds = 0.0003
delay_samples = int(delay_seconds * sampling_rate)

mic1_signal = source_signal
mic2_signal = np.pad(source_signal[:-delay_samples], (delay_samples, 0), mode='constant')

# Cross-correlation to calculate time delay
correlation = correlate(mic2_signal, mic1_signal, mode='full')
```

```
lags = np.arange(-len(mic1_signal) + 1, len(mic2_signal))
lag_index = np.argmax(correlation)
time_delay = lags[lag_index] / sampling_rate

# Estimate angle of arrival
try:
angle_rad = np.arcsin(time_delay * speed_of_sound / mic_distance)
angle_deg = np.degrees(angle_rad)
except ValueError:
angle_deg = None  # If value out of domain for arcsin

# Output results
print("Estimated Time Delay (TDOA): {:.6f} s".format(time_delay))
if angle_deg is not None:
print("Estimated Angle of Arrival: {:.2f}°".format(angle_deg))
else:
print("Error: Time delay too large to estimate angle (out of bounds)")

# Plot the signals and correlation (optional for report)
plt.figure(figsize=(12, 6))
plt.subplot(2, 1, 1)
plt.title("Mic Signals")
plt.plot(t, mic1_signal, label='Mic 1')
plt.plot(t, mic2_signal, label='Mic 2', alpha=0.7)
plt.xlabel("Time (s)")
plt.legend()

plt.subplot(2, 1, 2)
plt.title("Cross-Correlation")
plt.plot(lags / sampling_rate, correlation)
plt.xlabel("Lag (s)")
plt.tight_layout()
plt.show()
```

```
Estimated Time Delay (TDOA): 0.000292 s
Estimated Angle of Arrival: 30.01°
```



9.4 Backend Implementation

DeafReact > app > (tabs) > index.tsx > HomeScreen

```tsx
import React, { useState, useEffect } from "react";
import {
  View,
  Text,
  StyleSheet,
  Vibration,
  Dimensions,
  Animated,
  TouchableOpacity,
} from "react-native";
import { Client, Message } from "paho-mqtt";
import { MaterialCommunityIcons } from "@expo/vector-icons";
import { LinearGradient } from "expo-linear-gradient";
import * as Animatable from "react-native-animatable";

const HomeScreen = () => {
  const [hornValue, setHornValue] = useState<number>(0);
  const [timestamp, setTimestamp] = useState<number>(0);
  const [connectionStatus, setConnectionStatus] =
    useState<string>("Connecting...");
  const [alertActive, setAlertActive] = useState<boolean>(false);
  const [alertTimer, setAlertTimer] = useState<number>(0);
  const [lastTimestamp, setLastTimestamp] = useState<number>(0);
  const [client, setClient] = useState<Client | null>(null);
  const [orientation, setOrientation] = useState<string>(
    Dimensions.get("window").width > Dimensions.get("window").height
      ? "landscape"
      : "portrait"
  );
  const [windowWidth, setWindowWidth] = useState<number>(
    Dimensions.get("window").width
  );
  const [windowHeight, setWindowHeight] = useState<number>(
    Dimensions.get("window").height
  );
  const [radarRotation] = useState(new Animated.Value(0)); // For rada
  const [arrowScale] = useState(new Animated.Value(1)); // For popping
```

DeafReact > app > (tabs) > index.tsx > ...

```tsx
import React, { useState, useEffect } from "react";
import { View, Text, StyleSheet, Vibration } from "react-native";
import { Client, Message } from "paho-mqtt";
import { MaterialCommunityIcons } from "@expo/vector-icons";
import { LinearGradient } from "expo-linear-gradient";
import * as Animatable from "react-native-animatable";

const HomeScreen = () => {
  const [hornValue, setHornValue] = useState<number>(0);
  const [timestamp, setTimestamp] = useState<number>(0);
  const [connectionStatus, setConnectionStatus] = useState<string>("Connecting...");
  const [alertActive, setAlertActive] = useState<boolean>(false);
  const [alertTimer, setAlertTimer] = useState<number>(0); // Timer starts at 0, activates to 5 on detection
  const [lastTimestamp, setLastTimestamp] = useState<number>(0);
  const [client, setClient] = useState<Client | null>(null); // Properly typed as Client | null

  const connectToMQTT = () => {
    const mqttClient = new Client(
      "wss://e3b74a88eab74d1ea341225a79827748.s1.eu.hivemq.cloud:8884/mqtt",
      "react-native-" + Math.random().toString(36).substr(2, 9)
    );

    mqttClient.onConnectionLost = (responseObject) => {
      setConnectionStatus("Disconnected: " + responseObject.errorMessage);
      console.log("Connection lost:", responseObject.errorMessage);
      setTimeout(() => connectToMQTT(), 5000); // Retry after 5s
    };

    mqttClient.onMessageArrived = (message: Message) => {
```

DeafReact > app > (tabs) > index.tsx > HomeScreen

```tsx
const HomeScreen = () => {
  const renderLandscapeView = () => (
                <Text style={[styles.timer, { fontSize: scaleFont(8) }]}>
                  Alert: {alertTimer}s
                </Text>
              </Animatable.View>
            ) : (
              <View style={styles.placeholderSide}>
                <Text
                  style={[styles.placeholderText, { fontSize: scaleFont(12) }]}
                >
                  Left Side
                </Text>
              </View>
            )}
          </View>
          <View style={styles.splitSide}>
            <View style={styles.placeholderSide}>
              <Text
                style={[styles.placeholderText, { fontSize: scaleFont(12) }]}
              >
                Right Side
              </Text>
            </View>
          </View>
        </View>
        {!alertActive && (
          <View style={styles.radarOverlay}>
            <View style={styles.radarContainer}>
              <Animated.View
                style={[
                  styles.radarSweep,
                  { transform: [{ rotate: radarSpin }] },
                ]}
              />
              <Animatable.View
                animation="bounceIn" // Pops in when first rendered
```

esp32_microphone | Arduino IDE 2.3.4

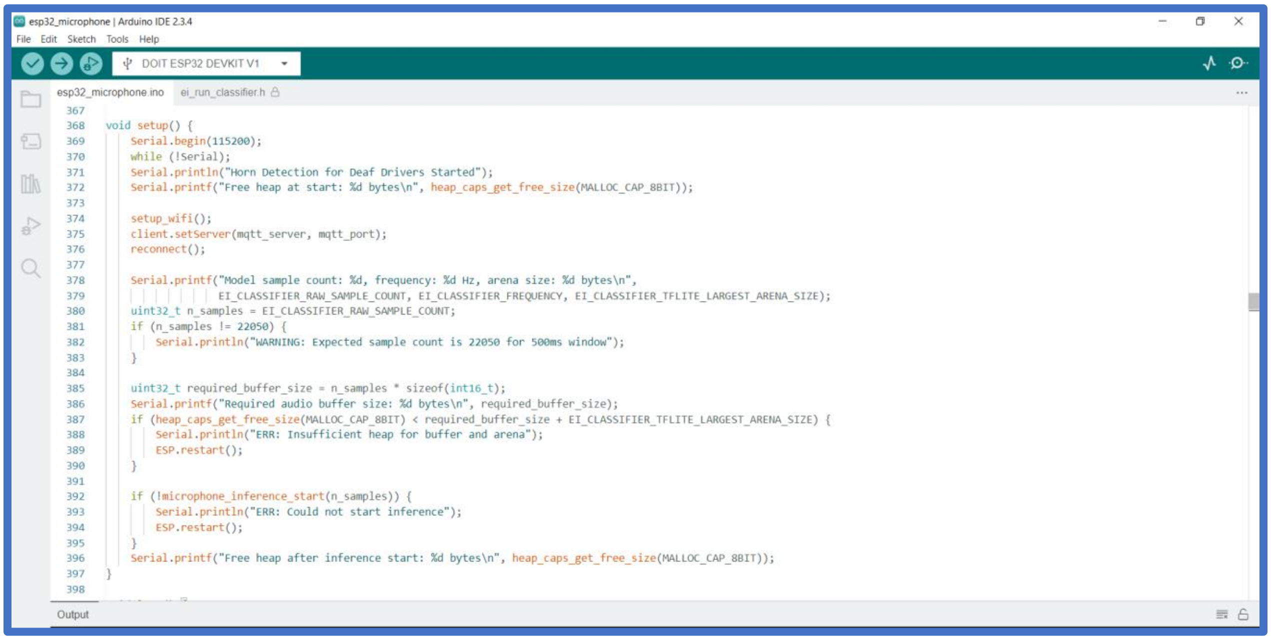
File Edit Sketch Tools Help

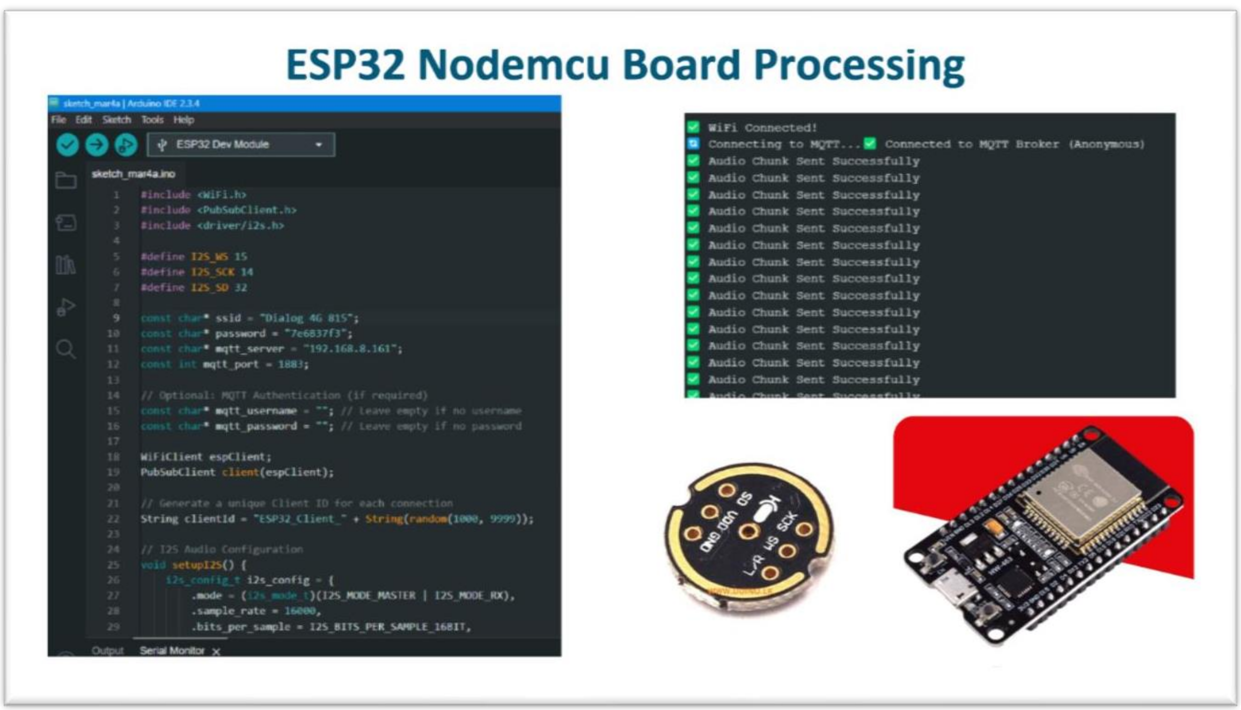
DOIT ESP32 DEVKIT V1

esp32_microphone.ino | ei_run_classifier.h

```cpp
      Serial.printf("Label: %s, Value: %.6f\n", result.classification[ix].label, result.classification[ix].value);
      if (strcmp(result.classification[ix].label, "Horn") == 0) {
        horn_value = result.classification[ix].value;
        Serial.printf("Horn value: %.6f\n", horn_value);
      }
    }
    if (horn_value > 0.85 && (millis() - last_detection_time > DETECTION_COOLDOWN)) {
      Serial.printf("HORN DETECTED: %.6f\n", horn_value);
      last_detection_time = millis();

      StaticJsonDocument<100> doc;
      doc["Horn"] = horn_value;
      doc["Timestamp"] = millis();
      char jsonBuffer[128];
      serializeJson(doc, jsonBuffer);

      int retries = 3;
      bool published = false;
      while (retries > 0 && !published) {
        if (client.publish(mqtt_topic, jsonBuffer)) {
          Serial.println("Prediction sent to MQTT");
          published = true;
        } else {
          Serial.printf("Failed to send prediction, rc=%d, retries left=%d\n", client.state(), retries);
          reconnect();
          delay(1000);
          retries--;
        }
      }
      if (!published) {
        Serial.println("ERR: Failed to publish after retries");
      }
```

Output


esp32_microphone | Arduino IDE 2.3.4
File Edit Sketch Tools Help
DOIT ESP32 DEVKIT V1
esp32_microphone.ino
ei_run_classifier.h
void setup() {
Serial.begin(115200);
while (!Serial);
Serial.println("Horn Detection for Deaf Drivers Started");
Serial.printf("Free heap at start: %d bytes\n", heap_caps_get_free_size(MALLOC_CAP_8BIT));
setup_wifi();
client.setServer(mqtt_server, mqtt_port);
reconnect();
Serial.printf("Model sample count: %d, frequency: %d Hz, arena size: %d bytes\n",
EI_CLASSIFIER_RAW_SAMPLE_COUNT, EI_CLASSIFIER_FREQUENCY, EI_CLASSIFIER_TFLITE_LARGEST_ARENA_SIZE);
uint32_t n_samples = EI_CLASSIFIER_RAW_SAMPLE_COUNT;
if (n_samples != 22050) {
Serial.println("WARNING: Expected sample count is 22050 for 500ms window");
}
uint32_t required_buffer_size = n_samples * sizeof(int16_t);
Serial.printf("Required audio buffer size: %d bytes\n", required_buffer_size);
if (heap_caps_get_free_size(MALLOC_CAP_8BIT) < required_buffer_size + EI_CLASSIFIER_TFLITE_LARGEST_ARENA_SIZE) {
Serial.println("ERR: Insufficient heap for buffer and arena");
ESP.restart();
}
if (!microphone_inference_start(n_samples)) {
Serial.println("ERR: Could not start inference");
ESP.restart();
}
Serial.printf("Free heap after inference start: %d bytes\n", heap_caps_get_free_size(MALLOC_CAP_8BIT));
}
Output


ESP32 Nodemcu Board Processing
sketch_mar4a | Arduino IDE 2.3.4
File Edit Sketch Tools Help
ESP32 Dev Module
sketch_mar4a.ino
#include <WiFi.h>
#include <PubSubClient.h>
#include <driver/i2s.h>
#define I2S_WS 15
#define I2S_SCK 14
#define I2S_SD 32
const char* ssid = "Dialog 4G 815";
const char* password = "7e6837f3";
const char* mqtt_server = "192.168.8.161";
const int mqtt_port = 1883;
// Optional: MQTT Authentication (if required)
const char* mqtt_username = ""; // Leave empty if no username
const char* mqtt_password = ""; // Leave empty if no password
WiFiClient espClient;
PubSubClient client(espClient);
// Generate a unique Client ID for each connection
String clientId = "ESP32_Client_" + String(random(1000, 9999));
// I2S Audio Configuration
void setupI2S() {
i2s_config_t i2s_config = {
.mode = (i2s_mode_t)(I2S_MODE_MASTER | I2S_MODE_RX),
.sample_rate = 16000,
.bits_per_sample = I2S_BITS_PER_SAMPLE_16BIT,
Output Serial Monitor
WiFi Connected!
Connecting to MQTT... Connected to MQTT Broker (Anonymous)
Audio Chunk Sent Successfully
SD
GND
VDD
WS
SCK
L/R

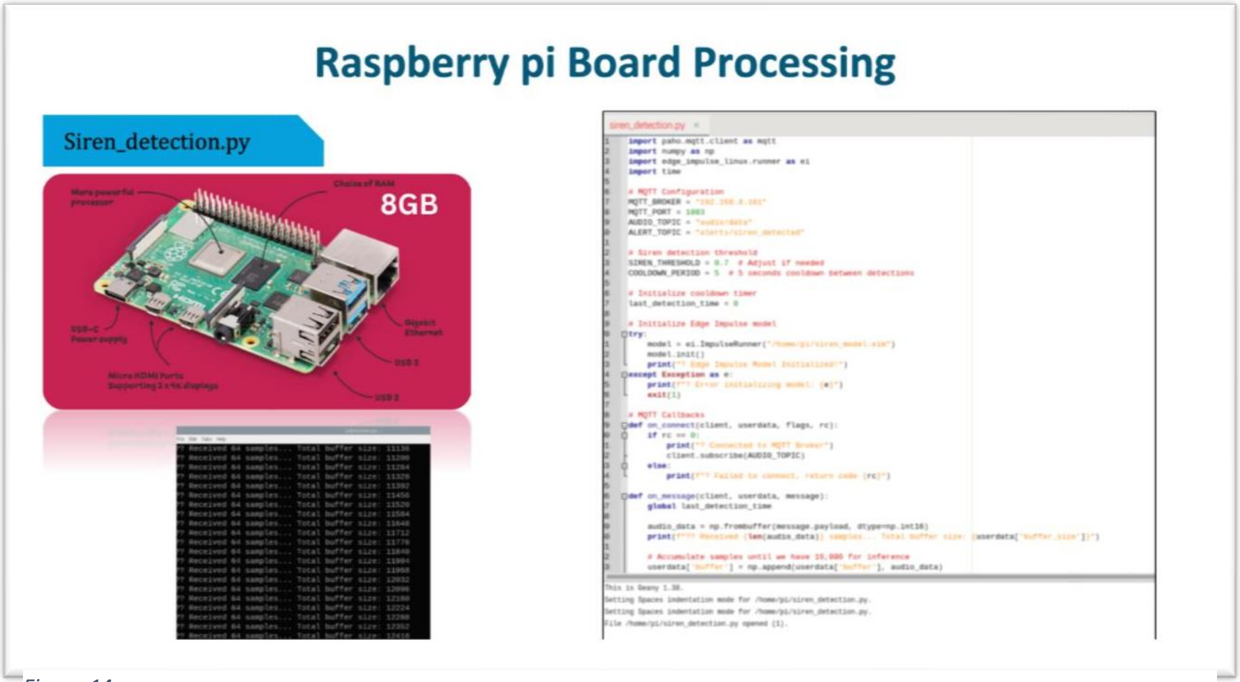
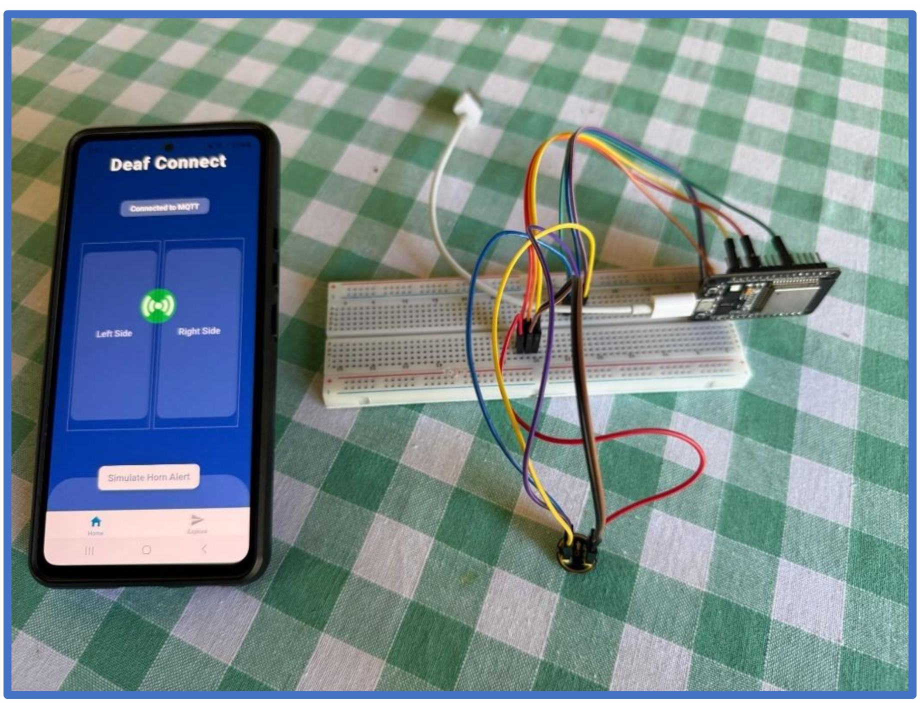
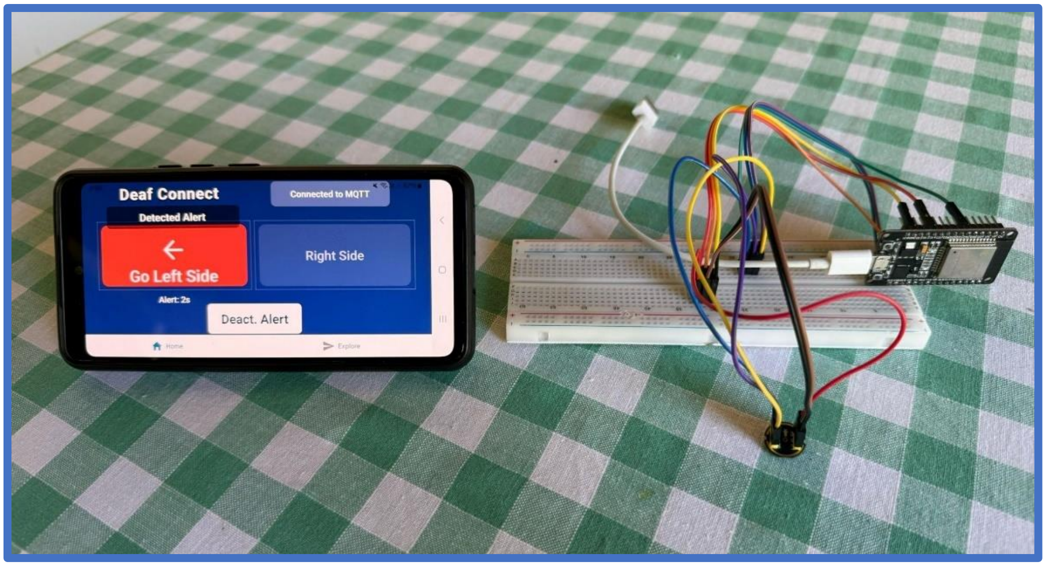
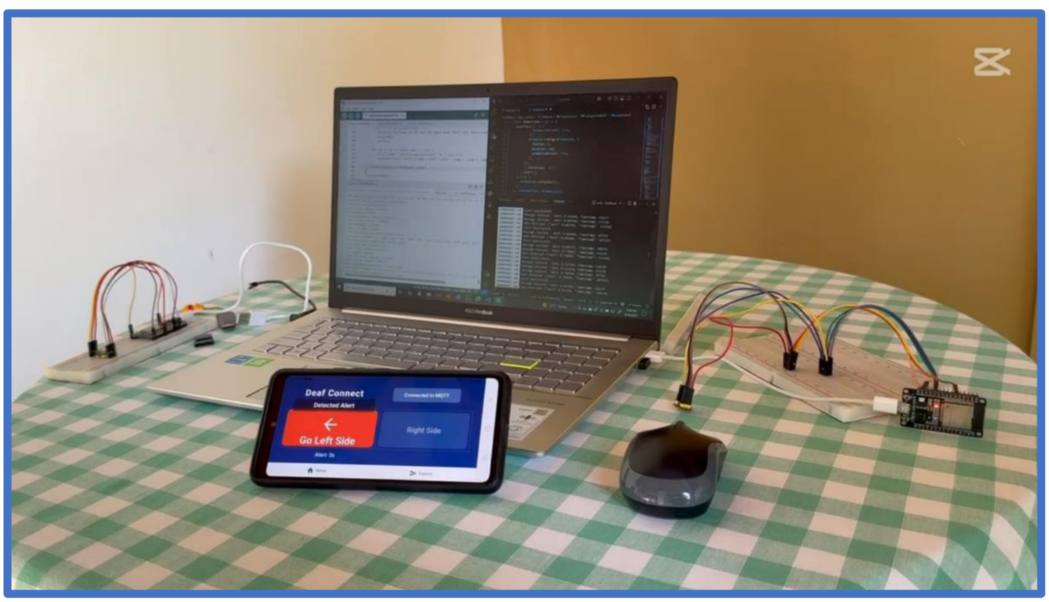


Figure 14

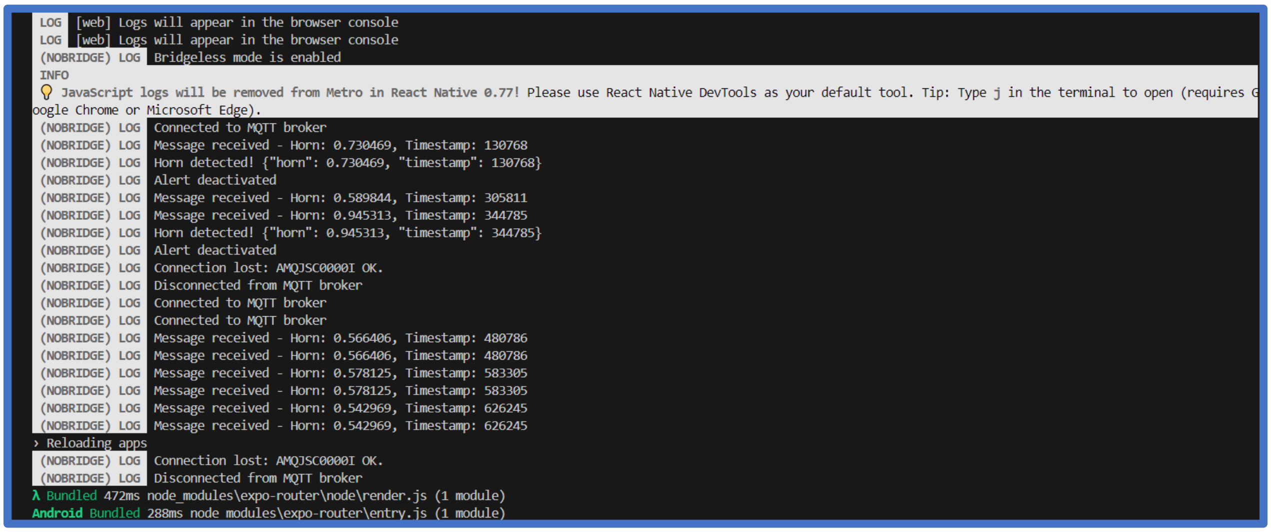
9.5 App Demo




Deaf Connect
Connected to MQTT
Detected Alert
Go Left Side
Right Side
Alert: 2s
Deact. Alert


Deaf Connect
Go Left Side

9.6 Testing Results



```
LOG  [web] Logs will appear in the browser console
LOG  [web] Logs will appear in the browser console
(NOBRIDGE) LOG  Bridgeless mode is enabled
INFO
💡 JavaScript logs will be removed from Metro in React Native 0.77! Please use React Native DevTools as your default tool. Tip: Type j in the terminal to open (requires Google Chrome or Microsoft Edge).
(NOBRIDGE) LOG  Connected to MQTT broker
(NOBRIDGE) LOG  Message received - Horn: 0.730469, Timestamp: 130768
(NOBRIDGE) LOG  Horn detected! {"horn": 0.730469, "timestamp": 130768}
(NOBRIDGE) LOG  Alert deactivated
(NOBRIDGE) LOG  Message received - Horn: 0.589844, Timestamp: 305811
(NOBRIDGE) LOG  Message received - Horn: 0.945313, Timestamp: 344785
(NOBRIDGE) LOG  Horn detected! {"horn": 0.945313, "timestamp": 344785}
(NOBRIDGE) LOG  Alert deactivated
(NOBRIDGE) LOG  Connection lost: AMQJSC0000I OK.
(NOBRIDGE) LOG  Disconnected from MQTT broker
(NOBRIDGE) LOG  Connected to MQTT broker
(NOBRIDGE) LOG  Connected to MQTT broker
(NOBRIDGE) LOG  Message received - Horn: 0.566406, Timestamp: 480786
(NOBRIDGE) LOG  Message received - Horn: 0.566406, Timestamp: 480786
(NOBRIDGE) LOG  Message received - Horn: 0.578125, Timestamp: 583305
(NOBRIDGE) LOG  Message received - Horn: 0.578125, Timestamp: 583305
(NOBRIDGE) LOG  Message received - Horn: 0.542969, Timestamp: 626245
(NOBRIDGE) LOG  Message received - Horn: 0.542969, Timestamp: 626245
› Reloading apps
(NOBRIDGE) LOG  Connection lost: AMQJSC0000I OK.
(NOBRIDGE) LOG  Disconnected from MQTT broker
λ Bundled 472ms node_modules\expo-router\node\render.js (1 module)
Android Bundled 288ms node_modules\expo-router\entry.js (1 module)
```

```
Label: Noise, Value: 0.804688
Free heap after classifier: 115212 bytes
Horn Detection for Deaf Drivers Started
Free heap at start: 235728 bytes
Connecting to WiFi........ connected!
Connecting to MQTT... connected
Model sample count: 8000, frequency: 16000 Hz, arena size: 13427 bytes
WARNING: Expected sample count is 22050 for 500ms window
Required audio buffer size: 16000 bytes
Free heap after inference start: 115584 bytes
Raw samples (first 16): -10496 -10512 -10464 -10496 -10480 -10464 -10496 -10480 -10480 -10496 -10464 -10480 -10464 -10464 -10448 -10464
Free heap before classifier: 115584 bytes
Classifier ran successfully
Label: Horn, Value: 0.042969
Horn value: 0.042969
Label: No Event , Value: 0.027344
Label: Noise, Value: 0.929688
Free heap after classifier: 115232 bytes
Raw samples (first 16): 16 0 -32 16 -32 -32 -96 -32 -64 -96 -96 -64 -64 -96 -128 -64
```

Output Serial Monitor ×

Not connected. Select a board and a port to connect automatically.

```
Free heap before classifier: 115080 bytes
Classifier ran successfully
Label: Horn, Value: 0.007812
Horn value: 0.007812
Label: No Event , Value: 0.949219
Label: Noise, Value: 0.042969
Free heap after classifier: 115080 bytes
Raw samples (first 16): -16 -64 0 -16 -32 0 -64 -32 -48 -64 -16 -48 16 -16 80 16
Free heap before classifier: 115080 bytes
Classifier ran successfully
Label: Horn, Value: 0.117188
Horn value: 0.117188
Label: No Event , Value: 0.343750
Label: Noise, Value: 0.539062
Free heap after classifier: 115080 bytes
Raw samples (first 16): 48 48 32 48 64 32 96 64 96 96 96 96 112 96 128 112
Free heap before classifier: 115080 bytes
Classifier ran successfully
Label: Horn, Value: 0.226562
Horn value: 0.226562
Label: No Event , Value: 0.269531
Label: Noise, Value: 0.503906
Free heap after classifier: 115080 bytes
Raw samples (first 16): -48 -48 -32 -48 -48 -32 -96 -48 -96 -96 -80 -96 -80 -80 -80 -80
Free heap before classifier: 115080 bytes
Classifier ran successfully
Label: Horn, Value: 0.148438
```

```
Label: No Event , Value: 0.066406
Label: Noise, Value: 0.765625
Free heap after classifier: 115080 bytes
Raw samples (first 16): -96 -80 -80 -96 -64 -80 -96 -64 -64 -96 -80 -64 -64 -80 -96 -64
Free heap before classifier: 115080 bytes
Classifier ran successfully
Label: Horn, Value: 0.242188
Horn value: 0.242188
Label: No Event , Value: 0.042969
Label: Noise, Value: 0.714844
Free heap after classifier: 115080 bytes
Raw samples (first 16): -176 -240 -160 -176 -128 -160 -128 -128 -80 -128 -48 -80 -8 -48 48 -8
Free heap before classifier: 115080 bytes
Classifier ran successfully
Label: Horn, Value: 0.003906
Horn value: 0.003906
```

```
Label: No Event , Value: 0.003906
Label: Noise, Value: 0.984375
Free heap after classifier: 115296 bytes
Raw samples (first 16): 6928 13568 -5376 6928 13376 -5376 -3296 13376 3472 -3296 -32 3472 3504 -32 -9568 3504
Free heap before classifier: 114976 bytes
Classifier ran successfully
Label: Horn, Value: 0.730469
Horn value: 0.730469
Label: No Event , Value: 0.000000
Label: Noise, Value: 0.269531
HORN DETECTED: 0.730469
Prediction sent to MQTT
Free heap after classifier: 113456 bytes
Raw samples (first 16): 96 -272 -784 96 -1056 -784 -352 -1056 64 -352 1264 64 -112 1264 -240 -112
Free heap before classifier: 115296 bytes
```

9.7 Dataset Descriptions

Horn Detection: 400 WAV files (50 local, 350 Edge Impulse), 2 s clips, 16 kHz.

LipCam: GRID corpus, 33,000 utterances, 6-token sentences.

Behavior Monitoring: 10,000 images (10 classes), 5,000 sensor sequences.

9.8 Code Snippets

Horn Detection: CNN and TDOA algorithms (Python, TensorFlow).

LipCam: CNN+GRU+CTC model (Python, Keras).

Behavior Monitoring: MobileNetV2 and LSTM fusion (Python, PyTorch).

9.9 Testing Protocols

Horn Detection: 50 trials, Galle Road, 85 dB noise.

LipCam: 5 speakers, lab and field (50–500 lux).

Behavior Monitoring: Simulated driving, 10 behaviors, 20% test split.

9.10 User Feedback Summaries

Ratings: 4.7/5 average, with haptic feedback rated highest (4.9/5).

Comments: “Vibrations felt natural, like a friend tapping me” (Volunteer A); “Texting alerts caught me every time” (Volunteer B).