

Sound Event Classification with TinyML on ESP32-S3 for Risk Alerting of Hearing-Impaired People in Public Spaces

Eduardo F. Parga Carvalho, Samuel de Jesus Monteiro de Barros, Rigel P. Fernandes^{ID} and Thiago Silva de Souza

Abstract—This paper presents a TinyML embedded system for environmental sound classification targeting hearing-impaired individuals in public spaces. A lightweight 1D Convolutional Neural Network (1D CNN) trained through the Edge Impulse platform is deployed on an ESP32-S3 microcontroller directly connected to a MAX9814 electret microphone, classifying three acoustic classes — vehicle horn, ambulance siren, and speech — entirely on-device without internet connectivity. Background silence is handled through a firmware-level amplitude threshold filter, bypassing unnecessary neural inference during acoustically inactive intervals and reducing processor workload. Experimental evaluation on physical hardware achieved 95.9% validation accuracy with 5 ms inference latency, confirming the feasibility of low-latency embedded audio classification for assistive safety applications. A BLE 5.0 haptic wearable is proposed as the next development stage.

Keywords—TinyML, sound event detection, hearing impairment, ESP32-S3, Edge Impulse, embedded classification.

I. INTRODUCTION

According to the IBGE, approximately 10.4 million Brazilians possess some degree of hearing impairment [1]. A significant portion frequents complex public spaces (subway stations, shopping centers, bus terminals, and busy sidewalks) where auditory risk signals play a critical safety role. Ambulance sirens, fire alarms, and vehicle horns are designed to warn pedestrians of imminent danger; however, individuals with hearing disabilities are systematically excluded from this crucial contextual information channel.

Existing assistive solutions fail to simultaneously satisfy three core architectural requirements: *autonomy* (operation without user intervention), *mobility* (continuous support during dynamic movement), and *intelligence* (the ability to distinguish specific risk sounds from background noise). Smartphone applications require active visual interaction and an unlocked screen, which may become impractical during emergencies. Fixed visual alert systems operate only at static locations, while traditional hearing amplifiers do not adequately support users with severe hearing loss [3], [4].

This work presents a TinyML-based embedded prototype capable of classifying environmental sounds directly on edge hardware without cloud dependency, thereby eliminating network-induced latency bottlenecks [4]. The proposed system

employs an ESP32-S3 microcontroller running a quantized neural model generated through Edge Impulse and deployed via TensorFlow Lite for Microcontrollers (TFLM) [8]. A wearable haptic notification module communicating through Bluetooth Low Energy (BLE 5.0) is proposed as future work. Related systems such as SSIES [2] and Liu et al. [3] validate the feasibility of embedded assistive sound recognition; however, this work emphasizes direct deployment and evaluation on the accessible ESP32-S3 hardware using the Edge Impulse pipeline, tailored for Brazilian urban accessibility.

II. SYSTEM ARCHITECTURE

The proposed system comprises two interconnected layers: the implemented embedded inference node responsible for acoustic processing, and a future wearable haptic notification module for tactile user feedback.

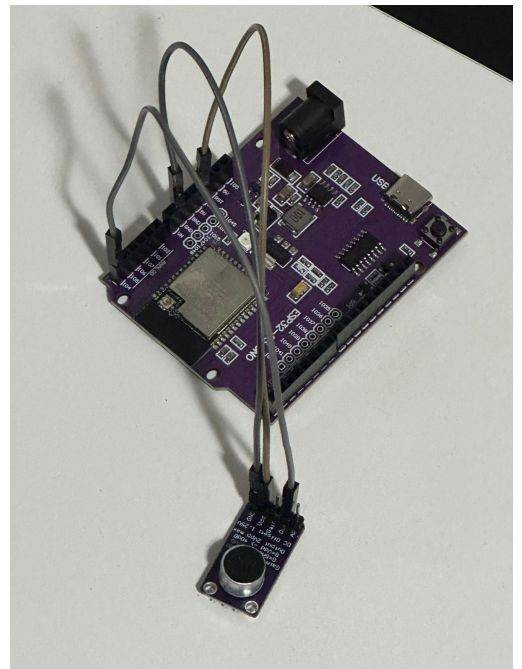


Fig. 1. Physical prototype: ESP32-S3 development board directly connected to the MAX9814 microphone module.

Eduardo F. Parga Carvalho, Samuel de Jesus Monteiro de Barros, Rigel P. Fernandes, and Thiago Silva de Souza are affiliated with the Tech Programs at Centro Universitário IBMEC, Rio de Janeiro, Brazil. Emails: edu.f.parga@gmail.com, samuel.barros@ibmec.edu.br, rigelfernandes@gmail.com, t.souza@ibmec.edu.br.

A. Sensor Node Implementation

The inference node consists of an ESP32-S3 microcontroller directly wired to a MAX9814 electret microphone amplifier.

Direct point-to-point wiring was intentionally adopted to reduce contact instability and minimize high-frequency analog noise commonly introduced by temporary breadboard matrix connections. The MAX9814 features a built-in Automatic Gain Control (AGC) module, outputting a stable analog audio signal sampled by the ESP32-S3 integrated analog-to-digital converter (ADC) at a steady 16 kHz rate with 12-bit resolution.

The firmware implements a highly optimized dual-stage processing pipeline. Initially, a lightweight amplitude threshold filter continuously monitors the peak-to-peak signal envelope. If the acoustic amplitude remains below a calibrated hardware threshold value of 2100, the system categorizes the environment as silent or background noise, bypassing neural inference execution entirely. This baseline rejection strategy drastically reduces unnecessary computations, lowering processor workload and preserving energy during quiet intervals. Once the predefined threshold is exceeded, the captured audio buffer is normalized and forwarded to the TFLM runtime.

The embedded model classifies three acoustic categories: vehicle horn, ambulance siren, and speech. The speech class functions exclusively as a non-alerting active rejection layer, improving robustness against false positives triggered by nearby human conversation in noisy public environments [9]. An 80% confidence threshold is additionally enforced during inference to suppress weak predictions.

B. Haptic Wearable (Proposed Next Stage)

The proposed wearable module will receive the inferred class identifiers from the sensor node via BLE 5.0 GATT notifications. A secondary ESP32 microcontroller will process these messages to activate a Linear Resonant Actuator (LRA) using a DRV2605L haptic driver. Distinct vibration patterns are planned for each risk category (Table I), enabling intuitive tactile awareness without requiring continuous visual attention or smartphone handling.

TABLE I. Proposed haptic feedback mapping.

Sound Event	Vibration Pattern	Urgency
Vehicle horn	1 medium intense pulse	High
Ambulance siren	3 long pulses (repeat)	Critical
Fire alarm *	5 rapid short pulses	Critical

* planned for future model expansion

III. TINYML MODEL AND TRAINING PIPELINE

A. Dataset Specification and Augmentation

The training dataset was constructed using carefully curated source audio files organized into three labeled classes: vehicle horn, ambulance siren, and speech. All recordings were processed within the Edge Impulse platform [5], resampled to a standardized 16 kHz mono WAV format, and segmented into uniform 1-second analytical windows.

Extensive data augmentation techniques were applied within the ingestion pipeline to prevent model overfitting and simulate real-world acoustic conditions. These techniques included

temporal shifting (± 200 ms), continuous amplitude variation ($0.5\times$ to $1.5\times$), and Gaussian white noise injection. The augmented corpus was divided using a strict 80/20 train-validation split.

B. Feature Extraction and Neural Architecture

Feature extraction is performed through a Mel-filterbank energy (MFE) processing block computing 40 Mel-frequency coefficients from each 1-second audio window using a 256-point Fast Fourier Transform (FFT) and a 20 ms frame stride. The extracted feature tensor is processed by a lightweight 1D Convolutional Neural Network (CNN) generated through Edge Impulse, structured as follows:

- Input layer: 3960 extracted MFE features;
- Reshape layer: 40 columns;
- 1D conv layer: 8 filters, kernel size 3;
- Dropout: rate 0.25;
- 1D conv layer: 16 filters, kernel size 3;
- Dropout: rate 0.25;
- Flatten layer;
- Output Dense layer: softmax, 3 classes.

The trained network was int8-quantized and exported as a TFLM C++ Arduino library. Edge Impulse's EON Compiler minimized memory consumption while preserving classification accuracy [6]. Training configuration is detailed in Table II. The code is available at [10]

TABLE II. Training and model configuration parameters.

Parameter	Value
Audio Input	1-second, 16 kHz mono
Feature Extraction	MFE (40 Mel bands)
FFT Size / Frame Stride	256 / 20 ms
Neural Architecture	1D CNN
Optimizer / Loss	Adam / Cross-Entropy
Training Epochs	100
Learning Rate	0.005
Quantization	Post-Training int8
Output Classes	3

IV. EXPERIMENTAL RESULTS AND DEPLOYMENT

The physical embedded prototype successfully performs on-device sound classification across three classes. Table III summarizes the validation metrics obtained from the deployed model on the ESP32-S3 hardware.

The per-class confusion matrix (Table IV) demonstrates strong categorical separation. Vehicle horn and speech achieved perfect 100% accuracy. Ambulance siren reached 91.8%, with residual confusion attributable to spectral overlap with the horn class under real-world acoustic conditions.

Real-time physical execution was validated through continuous observation of the classifier output during embedded operation (Fig. 2). The system achieved stable prediction transitions with an average inference latency of 5 ms — a direct consequence of the 1D CNN architecture operating over sequential MFE vectors rather than heavier 2D spectrogram

TABLE III. On-device execution and validation metrics.

Metric	Value
Validation Accuracy	95.9%
Validation Loss	0.13
Weighted F1-Score	0.96
Weighted Precision	0.96
Weighted Recall	0.96
Area Under ROC Curve (AUC)	1.00
Average Inference Latency	5 ms
Peak RAM Usage	14.7 KB
Flash Memory Usage	41.2 KB

TABLE IV. Validation confusion matrix (% true positives per class).

	Horn	Siren	Speech	F1
Horn	100%	0%	0%	0.94
Siren	6.6%	91.8%	1.6%	0.96
Speech	0%	0%	100%	0.92

representations. This efficiency allows the ESP32-S3 to sustain continuous audio sampling within a minimal footprint of 14.7 KB RAM and 41.2 KB flash.

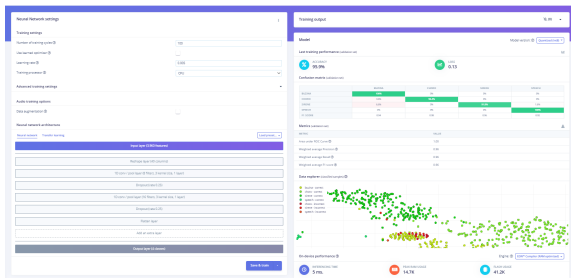


Fig. 2. Edge Impulse classifier output validating real-time prediction accuracy across the three implemented classes.

From an accessibility perspective, the proposed wearable architecture offers a critical non-visual awareness mechanism. The firmware-level silence filter (threshold 2100) contributes significantly to overall system efficiency, preventing inference loops during acoustically inactive intervals.

V. CONCLUSIONS AND FUTURE WORK

This paper presented a TinyML-based embedded environmental sound classification system implemented on the ESP32-S3 microcontroller, targeting assistive risk awareness applications for hearing-impaired individuals. The proposed architecture combines a firmware-level baseline rejection mechanism with a lightweight 1D CNN classifying three acoustic classes — vehicle horn, ambulance siren, and speech. Experimental validations confirmed 95.9% accuracy with an ultra-low 5 ms inference latency and a negligible memory footprint, validating the deployment of TinyML for real-time assistive safety technology [4].

Future development stages include the physical integration of the BLE 5.0 haptic wearable module and extensive evaluations under overlapping urban noise conditions. The

classification vocabulary will be further expanded to encompass additional high-urgency environmental alerts, evolving the standalone sensor node into a comprehensive, multi-tiered safety ecosystem.

REFERENCES

- [1] IBGE, “Censo Demográfico 2022: Pessoas com deficiência,” 2022. [Online]. Available: <https://www.ibge.gov.br>
- [2] C. Bautista-Villalpando and V. Lomas-Barrie, “A TinyML device for risk identification for people with hearing loss,” *Internet of Things*, Elsevier, vol. 35, p. 101840, Jan. 2026, doi: 10.1016/j.iot.2025.101840.
- [3] S.-H. Liu et al., “A Wearable Assistant Device for the Hearing Impaired to Recognize Emergency Vehicle Sirens with Edge Computing,” *Sensors*, MDPI, vol. 23, no. 17, p. 7454, 2023, doi: 10.3390/s23177454.
- [4] S. Heydari and Q. H. Mahmoud, “Tiny Machine Learning and On-Device Inference: A Survey of Applications, Challenges, and Future Directions,” *Sensors*, MDPI, vol. 25, no. 10, p. 3191, 2025, doi: 10.3390/s25103191.
- [5] Edge Impulse Inc., “Edge Impulse Documentation: Audio Classification,” 2025. [Online]. Available: <https://docs.edgeimpulse.com>
- [6] M. González et al., “Deploying Real-Time Speech Recognition on ESP32 Using TinyML and Edge Impulse,” *Springer CCIS*, vol. 2552, 2025, doi: 10.1007/978-3-031-97907-1_17.
- [7] R. David et al., “TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems,” *arXiv preprint arXiv:2010.08678*, 2020, doi: 10.48550/arXiv.2010.08678.
- [8] P. Warden and D. Situnayake, *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. O’Reilly Media, 2020, ISBN: 9781492052043.
- [9] A. Elhanashi et al., “Advancements in TinyML: Applications, Limitations, and Impact on IoT Devices,” *Electronics*, MDPI, vol. 13, no. 17, p. 3562, 2024, doi: 10.3390/electronics13173562.
- [10] E. Parga, “TCC - Sound Event Classification with TinyML,” *GitHub Repository*, 2026, disponível em: <https://github.com/EduardoParga/TCC---Sound-Event-Classification-with-TinyML->.