University of Pisa

Master Of Science in Computer Engineering

Industrial Applications

Guardian

Professors:

Pierfrancesco Foglia

Antonio Cosimo Prete

Students:

Carlo Pio Pace

Prashant Gautam

Academic Year 2025-2026

Abstract

Preventable fatalities from vehicular heatstroke continue to occur globally, resulting from instances where children or pets are inadvertently left by themselves or where children gain unsupervised access to vehicles. To address this safety challenge, we propose a Child and Pet Presence Detection (CPPD) system, covering its architecture, deployment, and performance assessment. This system employs visual and acoustic sensors to acquire data processed by a Raspberry Pi 3B+ to actively monitor the cabin environment immediately after vehicle closure. Upon detecting a living occupant, the system sends alerts to a designated caregiver. A functional prototype was developed to demonstrate the system's architectural logic and hardware integration. A performance evaluation was conducted to assess the system's detection response time. The results indicate that the system achieves reasonable detection accuracy for a prototype implementation, demonstrating its viability as a promising safety feature for modern automotive design.

Table of Contents

1 Introduction

The phenomenon of *vehicular heatstroke*, often referred to as "Hot Car Death," is a critical safety issue that continues to claim the lives of vulnerable occupants, particularly young children and pets. When a vehicle is parked in direct sunlight, the interior temperature can rise rapidly due to the greenhouse effect. Research indicates that within just 10 minutes, the internal temperature can increase by 20°F (approx. 11°C), reaching fatal levels even on days with mild ambient temperatures. Despite public awareness campaigns, these incidents frequently occur due to "Forgotten Baby Syndrome," a psychological phenomenon where a parent or caregiver inadvertently leaves a quiet child or a pet in the rear seat due to a lapse in routine, fatigue, or distraction. While child fatalities often receive significant media attention, similar incidents involving pets are frequently underreported, leading to an underestimation of the true scale of this global issue. As modern vehicles become increasingly soundproof and feature darker window tints, manual detection by passersby is rendered difficult. This necessitates an automated technological solution to mitigate these risks and prevent such fatalities.

1.1 Problem Statement

Current automotive safety standards are evolving, but many existing solutions are limited to high-end luxury vehicles or rely on passive reminders (such as "check rear seat" messages) that can be easily ignored or habituated by the driver. There is a significant need for an active, cost-effective detection system that can autonomously sense the presence of a living being after the driver exit from the car and take immediate preventive action.

1.2 Related works

Child Presence Detection (CPD) has attracted increasing attention from both academia and industry due to its life-saving potential and the growing regulatory requirements for in-vehicle safety systems. Existing CPD solutions mainly rely on sensing technologies such as wireless radio signals, vision-based systems, and more recently, thermal and acoustic sensing. This section reviews representative works closely related to our study.

Thermal imaging-based CPD: Vision-based CPD systems commonly rely on visible cameras; however, they are limited in low-light conditions and cannot directly assess thermal risks. To address this limitation, Choi *et al.* investigated the feasibility of an infrared thermal imaging-based CPD system [1]. Their proposed architecture enables simultaneous object detection, recognition, and status assessment, while also monitoring the thermal environment inside the vehicle. Unlike radar or camera-based solutions that primarily focus on presence or motion, thermal imaging provides additional information related to temperature, making it suitable for detecting hazardous situations such as heatstroke. Experimental results demonstrate that thermal imaging-based CPD can meet performance requirements specified by automotive regulations, although the system requires dedicated thermal sensors.

Radar sensing based CPD: *Gillner et al.* [2] details the design and implementation of a 24 GHz Continuous Wave (CW) radar system specifically engineered to detect the subtle chest movements associated with an infant's breathing. The authors propose a cost-effective and highly integrated hardware architecture, utilizing a custom 4x1 patch array antenna and an 8-bit microcontroller with a dedicated hardware filter. By focusing on respiratory rate (RR) detection rather than simple motion or weight, the system aims to reliably identify sleeping infants who might otherwise be missed by traditional sensors. The paper concludes with a validation of the system's efficacy, demonstrating its ability to accurately monitor a sleeping child's respiratory patterns in real-time against established reference standards. *Nazri et al.* [3] explores the application of high frequency millimeter wave technology to enhance cabin safety.The authors propose a method utilizing multi-channel 60 GHz Frequency Modulated Continuous Wave (FMCW) radar. Unlike lower-frequency alternatives, the 60 GHz band offers superior resolution, allowing the system to generate detailed range-angle maps. The research focuses on applying clutter suppression techniques to eliminate noise from stationary vehicle interiors (such as seats) and employs machine learning classification algorithms—including Support Vector Machines (SVM) and Convolutional Neural Networks (CNN) to accurately identify passenger location and count. The study demonstrates that this radar-based approach can achieve high detection accuracy, robustly identifying children even when they are covered or motionless.

Acoustic sensing based CPD: Acoustic sensing has recently emerged as a promising modality for fine-grained human sensing. Su *et al.* introduced the concept of distributed acoustic sensing in car cabins by exploiting the dense deployment of built-in speakers and microphones [4]. Their system leverages the rich multipath characteristics of the enclosed cabin environment to detect child presence even when the direct reflection path is blocked. Using child presence detection as a case study, the authors demonstrate that both body movements and subtle breathing signals can be reliably captured across the entire cabin without blind spots, achieving high detection accuracy and low false alarm rates under challenging scenarios such as rear-facing seat blockage. This work highlights the potential of acoustic sensing for robust in-cabin CPD without requiring additional dedicated hardware.

In summary, prior works demonstrate that wireless, acoustic, and thermal sensing modalities are all viable approaches for CPD, each with distinct advantages and limitations. Wireless and acoustic solutions benefit from contactless sensing and wide coverage and privacy, while thermal imaging provides critical environmental and physiological context. These studies collectively motivate further exploration of robust, scalable, and cost-effective CPD systems that can operate reliably under diverse real-world in-cabin.

1.3 Scope of the study

The primary objective of this project is to design and implement a Camera and Audio based Child and Pet Presence Detection (CPPD) system for vehicles. The specific goals are:

1. To accurately detect the presence of occupants inside a stationary vehicle using a vision based AI model, gathering images from a camera.
2. To enhance detection reliability by addressing the inherent Field of View (FOV) limitations of cameras through the integration of an audio based model.
3. In case of risk, sending an alert to a designated caregiver.
4. Develop and demonstrate a working prototype that validates the feasibility, reliability, and effectiveness of the proposed multimodal detection architecture.
5. Evaluate the current State of the Art of Object detection and audio classification models, for implementation within the system.
6. To evaluate and address the challenges of deploying such systems on resource-constrained devices..
7. Evaluate the response time of the system.

1.4 System Description and Highlight

The Child and Pet Presence Detection (CPPD) system is a multimodal safety solution, it integrates a camera and microphone to monitor the vehicle interior and analyze occupant presence. An image based deep learning model detects and classifies the occupants. The presence of an adult is considered as a safe scenario, whereas detection of a child or a pet alone is treated as a potential risk. To enhance reliability and reduce false negatives, due Field of View problem of cameras, blind spots, etc. an audio model analyzes microphone input for sounds such as crying or pet sounds. When an unsafe condition is confirmed, the system automatically sends an alert notification to a designated caregiver. The architecture emphasizes real-time processing, camera and microphone sensing, and automated emergency response to improve in vehicle safety conditions.

2 State of the Art

2.1 Object Detection

Object detection remains a cornerstone of computer vision, particularly for safety-critical applications like child and pet presence detection. Recent advancements have seen a transition from traditional sliding-window methods to deep learning architectures that leverage Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs). According to a 2025 survey by Cao et al. [5], the integration of ViTs has significantly enhanced the ability of models to understand deep image properties and global context, which is essential for distinguishing small subjects in cluttered indoor environments.
For real-time monitoring, the YOLO framework continues to be the industry standard. Ali and Zhang [6] provide an extensive review of the YOLO lineage up to version 11, noting its efficiency in balancing inference speed with high mean Average Precision (mAP). This evolution reached a new milestone with the release of YOLO26 in late 2025. As detailed by Sapkota et al. [7], YOLO26 introduces NMS-free inference and Small-Target-Aware Label Assignment (STAL). These features are particularly relevant for detecting infants or small pets that might otherwise be overlooked by standard detectors due to their size or unconventional orientations.
However, deploying these systems on edge devices or mobile platforms requires a focus on "lightweight" architectures. Yerulan and Zhasdauren [8] conducted a comparative analysis of YOLOv8n, MobileNet-SSD, and NanoDet, highlighting the trade-offs between model size and error rates. While YOLO variants often lead in precision, anchor-free models like NanoDet offer reduced computational overhead. Furthering this efficiency, Van Thanh et al. [9] proposed an enhanced NanoDet variant utilizing an EfficientNetB0-compact backbone. This modification allows for superior performance metrics on mobile and edge computing environments, making it a highly viable candidate for low-power, "always-on" presence detection systems.

2.2 Audio Classification

Audio classification serves as a critical redundant modality in presence detection systems, identifying specific acoustic signatures such as infant cries or pet vocalizations. Recent research emphasizes the superiority of deep learning over traditional machine learning for handling non-stationary environmental noise. Gourisaria et al. [10] demonstrated that while traditional features like Mel-Frequency Cepstral Coefficients (MFCC) and Short-Time Fourier Transform (STFT) remain essential for signal representation, Artificial Neural Networks (ANN) significantly outperform other classifiers, achieving over 91% accuracy in complex industrial and safety

scenarios.
The current state of the art heavily favors transfer learning using pre-trained models optimized for edge deployment.YAMNet, a deep network trained on Google's AudioSet, has emerged as a primary benchmark. Valliappan et al. [11] utilized YAMNet for high-precision acoustic detection, reaching a 94.96% accuracy by leveraging Mel-spectrograms for feature extraction. This is further advanced by the YAMNet-Trans framework proposed by Liu et al. [12], which synergizes transfer learning with signal processing techniques to achieve 94.21% accuracy even with limited labeled data, making it ideal for specialized tasks like child or pet sound recognition in resource-constrained IoT environments.
For real-time monitoring where false positives can be costly, multi-representational encoding is a key trend. Podda et al. [13] introduced a pipeline utilizing intensity-projected spectrograms and custom CNNs to identify anomalous sounds, achieving a remarkably low false positive rate of 0.96%. Similarly, comparative evaluations by Sakaki et al. [14] between standard CNNs, VGG16, and Conformers underscore that for mobile-ready systems, models like YAMNet and its variants provide the most efficient balance of latency and discriminative power.

3 System Architecture

This section details the proposed architecture of the Child and Pet Presence Detection System. The system is designed as a multimodal IoT application, capable of processing visual and auditory data in real-time to detect vulnerable entities within a vehicle. The main components of the system are:

The Eye Module: This module acts as the visual pipeline: it ingests raw camera feeds, executes image pre-processing, sends the pre processed images to the image classification module.

The Ear Module: The Audio Processing Module monitors the acoustic environment via microphones. It captures and pre-processes audio segments, transmits them to the audio classification module.

Audio classification module : Upon receiving the data from the Ear module, run the inference, and forwards the result to the Brain module.

Image classification module : Once received the data from the Eye module, it performs the inference and transmits the results to the Brain.

Brain Module: Serving as the central orchestrator, this process synchronizes the 'Ear' and 'Eye' modules. It initiates data acquisition, wait for the inference results, aggregates them, and assesses the cabin occupants safety. If a critical risk is detected, it immediately activates the Communications Module.

The Communications module: Upon receiving a trigger from the Brain module, this component transmits an alert to the designated caregiver, which will receive the alert on his smartphone.
The diagram below depicts the parallel execution of the Eye and Ear processes, their synchronization via the Brain module, and the final alert transmission to the Telegram API.

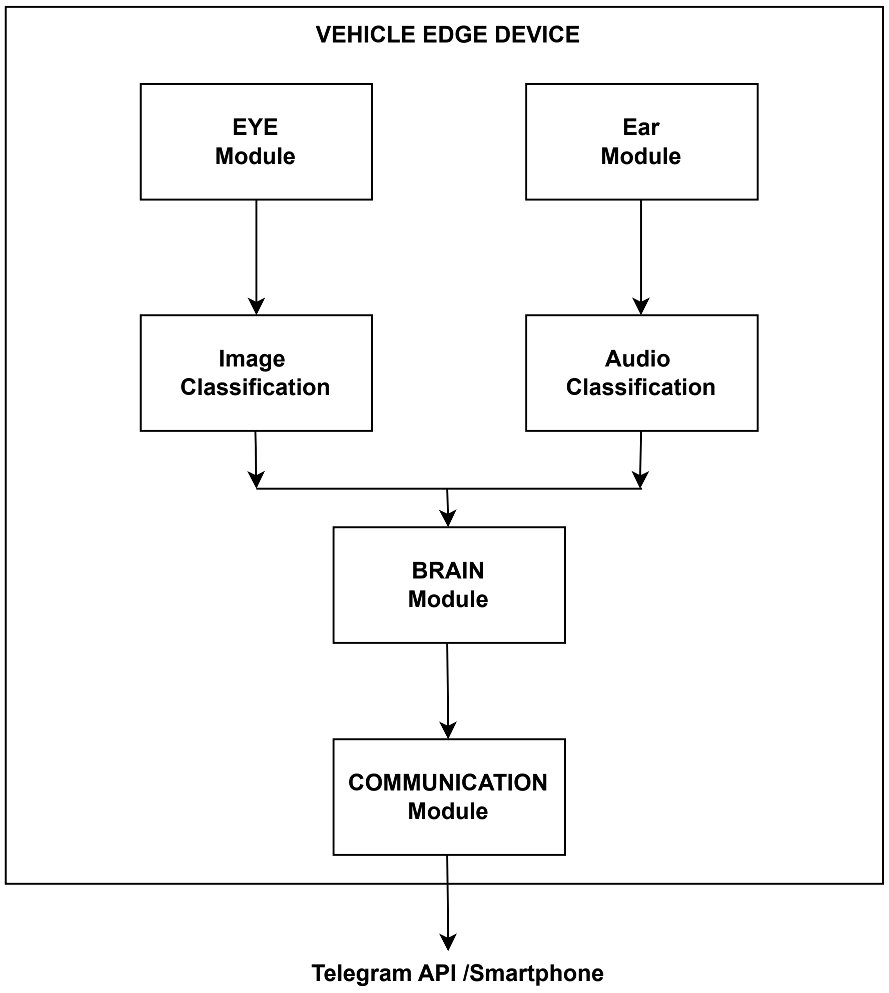


Figure 1: **High-Level System Architecture.**

4 AI Solutions Evaluation

To select the most appropriate AI models for the prototype, we conducted a preliminary evaluation of reliable, state of the art solutions designed for edge computing, previously mentioned in chapter 2.

4.1 Evaluation Method

The audio and image models were evaluated using custom datasets. It should be noted that the scale of these datasets is insufficient for commercial automotive deployment. Consequently, the results presented in this study should be interpreted within the context of these data limitations.
The visual dataset comprises 1,700 images across varying resolutions and scenarios. A significant portion depicts in-cabin environments featuring diverse camera angles and lighting conditions. The target classes are defined as follows:

- Adult
- Child : category which includes also babies
- Pet : referring to dogs (including different breeds)

The dataset is approximately balanced, with roughly 500 images per category. Additionally, 150 images of empty scenes are included to represent the background class.
The audio dataset comprises 750 labeled audio samples, each with a duration of 5 seconds. The target classes for the audio classification task are defined as follows:

- Adult sound
- Child sound
- Noise
- Pet sound (dog barks)
- Silence

Each category contains 150 audio samples, resulting in a balanced dataset of 750 total samples.
The Adult sounds category includes speech fragments, conversational tones, and other human generated sounds typical of adult occupants. The Child sound class specifically captures distress-related crying sounds, which are critical indicators for emergency detection scenarios. The Child sounds category also includes non-distress

vocalizations such as babbling, laughing, or playful sounds. The Pet sounds class refers to dog barking. The Noise category represents non-informative environmental sounds such as vehicle vibration, airflow, or external disturbances. Finally, Silence represents low-amplitude or near-silent cabin conditions, serving as a baseline reference for model calibration.

4.2 Image based Models Results

The results of the evaluation in table 1, shows an mAP50 metric pretty high considering, the small dataset, this is the result of the transfer learning, since all the listed model have been pretrained on the COCO dataset.

MODEL	**PARAMETERS [millions]**	**RESOLUTION**	**mAP50**
picoDET	2.15	416	0.86
nanoDET	6.1	480	0.89
Yolo v5 NU	2.6	640	0.924
Yolo v11 N	2.6	640	0.917
Yolo v12 N	2.6	640	0.922
Yolo v26 N	2.4	480	0.905
Yolo v26 N	2.4	640	0.928
Yolo v26 S	9.5	640	0.947

Table 1: Performance Comparison of Object Detection Models

Since the system must be deployed on an edge device with limited computational resources, the choice of model was guided by the safety critical nature of the application and the requirement for high-speed inference. We selected the YOLOv26-N model as it provides the optimal compromise between mAP50 and model size.

4.3 Audio based Models Results

The performance and complexity of the three audio classification architectures are tabulated in Table 2. The YAMNet+FFN is a custom solution in which we leverage the pre-trained YAMNet model as a feature extractor to feed a custom Feed-Forward Network trained on our specific dataset. This solution achieves the highest accuracy among all evaluated models.

MODEL	**PARAMETERS [millions]**	**ACCURACY**
Spectogram CNN	0.024	0.72
MFCC + SVM	0.05	0.86
YAMNet + FFN	0.66	0.88

Table 2: Performance Comparison of Audio Classification Models

The selection of YAMNet as our final Audio model was driven by two primary factors : the safety critical nature of the system and edge device effectiveness.

5 Prototype and Demo Setup

This section details the hardware architecture, software environment, and deployment strategies utilized in the Guardian prototype. Specifically, it outlines the implementation of video and audio models on the Raspberry Pi 3B+, emphasizing real-time performance and resource efficiency in an edge computing context.

5.1 Hardware Description

The hardware implementation relies on off the shelf components chosen for their compatibility and ease of integration. Below, we describe the specific sensors and processing unit used to construct the prototype.

5.1.1 Raspberry Pi 3 Model B+

The system is implemented using the Raspberry Pi 3 Model B+ (Rev 1.3) [15], which functions as the central processing and control unit. It is based on the ARMv8 64-bit architecture and powered by a quad-core ARM Cortex-A53 processor operating at up to 1.4 GHz. The board includes 1 GB LPDDR2 RAM and uses a microSD card as primary storage.
It provides built-in Wi-Fi, Bluetooth, Ethernet, USB ports, HDMI output, and a 40-pin GPIO header for hardware interfacing. In this system, the Raspberry Pi interfaces with the camera module via the CSI-2 port and manages image capture, processing, storage, and network transfer.

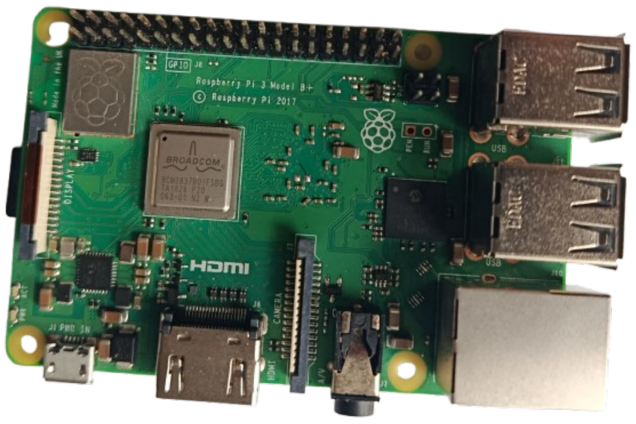


Figure 2: Raspberry Pi 3 Model B+

5.1.2 Flash Memory

The system utilized a 64 GB SanDisk Ultra microSDXC UHS-I flash memory card (model SDSQUAB-064G-GN6MA) as the primary non-volatile storage. [16].

Figure 3: Flash memory

5.1.3 Camera Module

The system uses the Raspberry Pi Camera Module v2 [17], based on the Sony IMX219 8-megapixel image sensor. The camera is connected to the Raspberry Pi through the MIPI CSI-2 (Camera Serial Interface), enabling high-speed image data transfer.
The sensor supports a maximum resolution of 3280 × 2464 pixels and multiple operating modes such as 640×480, 1640×1232, and 1920×1080. In this system, the camera is configured using the `libcamera2` framework and controlled through `rpicam` utilities for image capture.

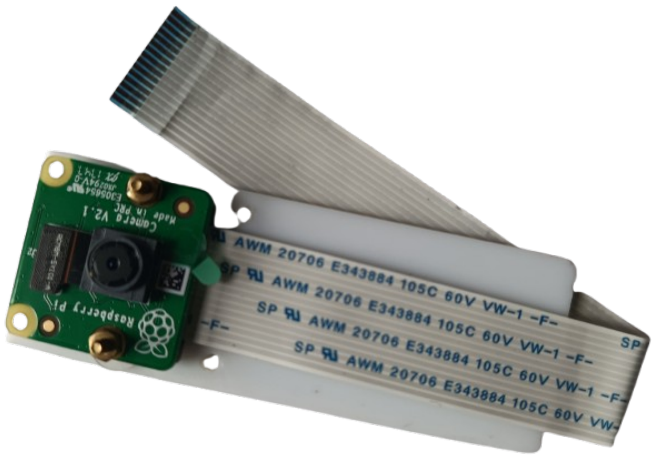


Figure 4: Raspberry Pi Camera Module v2

5.1.4 Microphone Module

Audio input in the system is provided through a USB Plug-and-Play (PnP) Sound Device connected to the Raspberry Pi via a USB interface. Since the Raspberry Pi 3 Model B+ does not include a built-in microphone, an external USB audio device is used for sound acquisition.
The microphone operates as a USB Audio Class (UAC) compliant device, enabling automatic driver support under Debian GNU/Linux without additional configuration. It supports real-time audio recording for signal processing and analysis tasks.

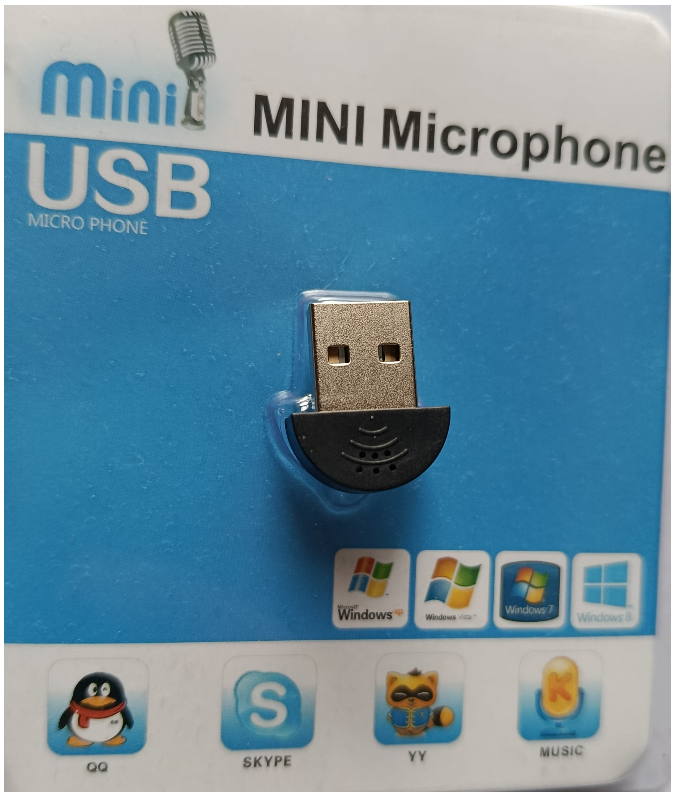


Figure 5: USB Plug-and-Play Microphone

Parameter	Details
Manufacturer	Mini
Model	MI-305
Sensitivity	6̃7 dBv/pBar, 4̃7 dBV/Pascal +/-4dB
Frequency response	100-16kHz

Table 3: Technical Specifications of microphone

5.2 Raspberry Pi 3B+ Configuration

The proposed system was deployed using a optimized software stack configured for edge processing. The technical specifications are described below.

5.2.1 Operating System Setup

To optimize the system for real-time child and pet detection, we utilized a streamlined operating system environment. By opting for the Raspberry Pi OS Lite (64-bit) based on Debian Bookworm, we effectively removed the overhead of a Graphical User Interface (GUI) and unnecessary desktop services.
The technical specifications extracted from the system are summarized in Table 4.

Parameter	Details
Operating System	Raspberry Pi OS Lite (64-bit)
Debian Version	12 (Bookworm)
Kernel Version	6.12.47+rpt-rpi-v8
Kernel Build	#1 SMP PREEMPT (2025-09-16)
Architecture	ARMv8 64-bit (aarch64)
Hostname	guardian
Userland	GNU/Linux
Optimization	Preemptive Kernel (Low-Latency)

Table 4: Software Environment and OS Specifications

5.2.2 Python Environment and Dependencies

The core software logic is implemented using Python 3.11.2, which provides a stable and optimized runtime for the Debian Bookworm environment. To maintain a clean system state and avoid dependency conflicts, all libraries were managed within a dedicated Python virtual environment (venv).The system utilizes a combination of high-level wrappers for hardware interfacing and low-level runtimes for model inference. The specific requirements used in the stable deployment are detailed in Table 5.

Library	**Version**	**Primary Role**
tflite_runtime	2.14.0	Optimized edge inference engine
cv2 (OpenCV)	4.6.0	Real-time image/video stream processing
picamera2	0.3.31	Libcamera-based high-level camera interface
scipy	1.17.0	Signal processing and mathematical routines
sounddevice	0.5.3	Real-time audio recording and playback
soundfile	0.13.1	Audio conversion library
numpy	1.24.2	High-performance array operations
requests	2.32.5	Cloud communication and notification API

Table 5: Python Dependencies and Versioning

5.2.3 Camera Integration

The camera was connected to Raspberry Pi using the ribbon cable inserted in the **MIPI CSI** connector (Figure 6). Image acquisition was performed using the `picamera2` library. The `Picamera2` class was utilized to capture raw frames, The settings used with picamera2 are 640×480 resolution, subsequently pre-processed with OpenCV to change from the BGR format (standard for picamera2) to RGB.

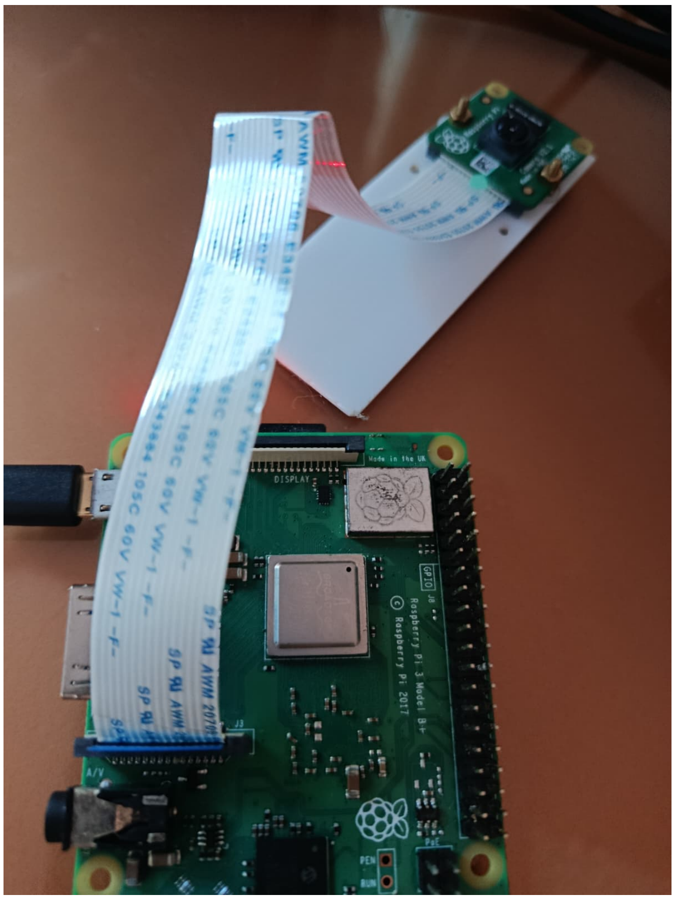


Figure 6: Camera integration

5.2.4 Microphone Integration

The microphone was connected to the Raspberry Pi using one the of the USB port available (Figure 7). Raw audio was captured utilizing the `Sounddevice` library. The sample rate utilized is 44100 Hz, using the command `alsamixer`, we increased the hardware gain by 62 points (Figure 8).

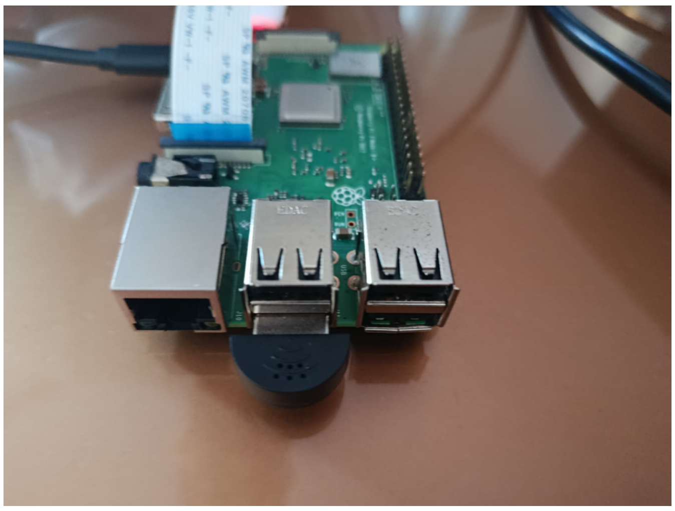
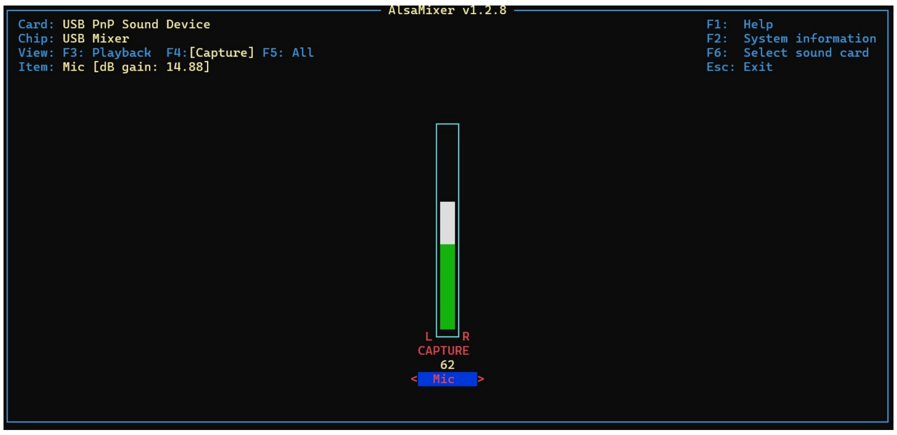


Figure 7: Microphone integration



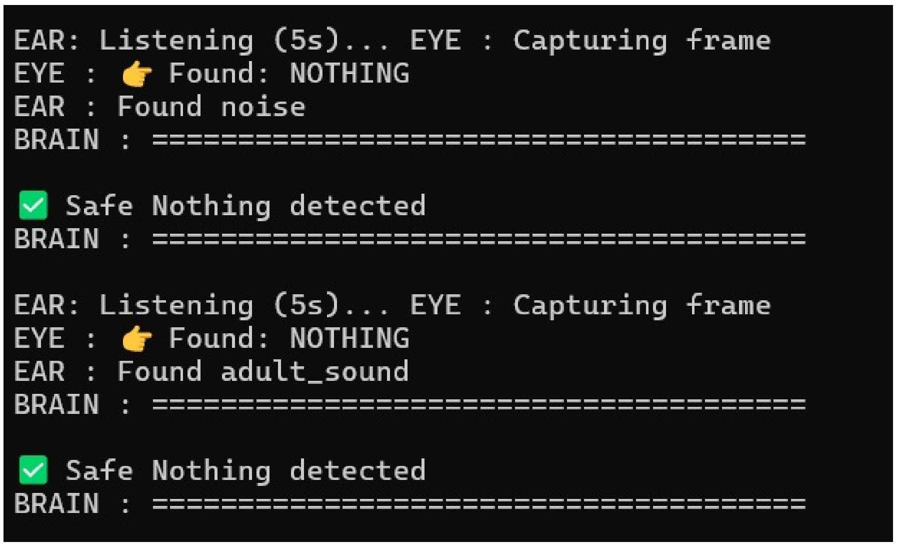
```
                                   AlsaMixer v1.2.8
Card: USB PnP Sound Device                                          F1:  Help
Chip: USB Mixer                                                     F2:  System information
View: F3: Playback  F4:[Capture] F5: All                            F6:  Select sound card
Item: Mic [dB gain: 14.88]                                          Esc: Exit

                                      L  R
                                    CAPTURE
                                       62
                                    < Mic >
```

Figure 8: Alsamixer settings

5.2.5 Output Examples

Figure 9 illustrates some of the possible outputs of the system.



```
EAR: Listening (5s)... EYE : Capturing frame
EYE : 👉 Found: NOTHING
EAR : Found noise
BRAIN : ======================================

✅ Safe Nothing detected
BRAIN : ======================================

EAR: Listening (5s)... EYE : Capturing frame
EYE : 👉 Found: NOTHING
EAR : Found adult_sound
BRAIN : ======================================

✅ Safe Nothing detected
BRAIN : ======================================
```

Figure 9: System Output: State evaluation based on audio and sound detections

5.3 Prototype

The primary objective of the proposed system is the robust detection and classification of vehicle cabin occupants. By utilizing the multimodal sensing capabilities (visual and acoustic) and classification categories mentioned in Section 4.1, the prototype is designed to mitigate environmental interference and accurately identify safety risks. The system operates based on two distinct operational states: :

1. **Safe state** : defined by the confirmed presence of an adult occupant via image or audio.

2. **Dangerous state** : triggered when a child or pet is detected in the absence of an adult. In this state, the system automatically transmits an alert to a designated caregiver.

5.3.1 Images pre-processing pipeline

The pre-processing involved resizing and padding (letterboxing) to preserve the aspect ratio of captured images, ensuring the input dimensions matched the requirements of the YOLO architecture. Additionally, `OpenCV` was employed to encode the images for transmission to designated caregivers upon alert generation.

5.3.2 Audio pre-processing pipeline

After the raw signal audio was acquired, using `Sounddevice` library, the pre-processing started flattening the signal using `Numpy`. To match the input requirements of the YAMNet architecture, with `Scipy` the signal was downsampled to 16 kHz using polyphase filtering. If necessary also the library `soundfile` was used to convert the raw signal into ".OGG" format, to send the audio alert.

5.3.3 Object Detection Model Deployment

To address the computational constraints of the Raspberry Pi hardware, the original YOLO26 architecture was converted to the TensorFlow Lite (TFLite) format. This conversion facilitates standalone inference, circumventing the high memory overhead associated with the `Ultralytics` library. Although model quantization was evaluated to further reduce latency, it was ultimately eschewed due to a significant degradation in detection accuracy. Consequently, the model was deployed using 32-bit floating-point precision (Float32) to maintain a robust balance between performance and precision. For the experimental evaluation, the Nano variants of YOLO26 were tested at both 480 × 480 and 640 × 640 input resolutions.

YOLO MODEL	INFERENCE TIME [s] (CI 99%)
Nano 480	0.8048 ± 0.0016
Nano 640	1.4343 ± 0.0056
Small 640	5.2359 ± 0.1177

Table 6: Inference time comparisons of Yolo Models

5.3.4 Audio Classification Model Deployment

While several architectures were evaluated, the YAMNet+FFN configuration was selected for the final implementation. This custom architecture utilizes a pre-trained YAMNet model as a feature extractor to leverage its extensive training on large-scale audio datasets. The resulting embeddings are then passed to a custom Feed-Forward Network (FFN), which serves as the classification head. For deployment optimization, both components were converted to the TensorFlow Lite (TFLite) format.

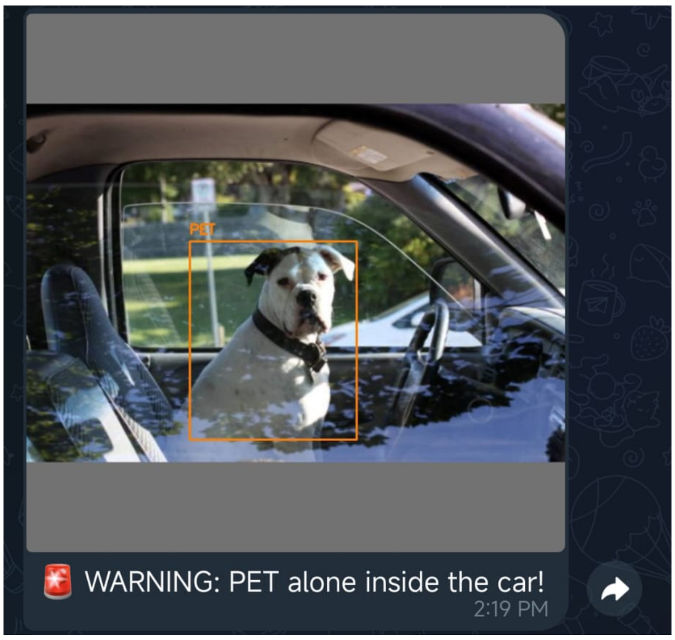
AUDIO MODELS	INFERENCE TIME [s] (CI 99%)
MFCC+SVM	0.0175 ± 0.003
CNN+SPECTROGRAM	0.0206 ± 0.005
YAMNet+FFN	0.02 ± 0.001

Table 7: Inference time comparisons of Audio Models

5.3.5 Notification System and Alert Modalities

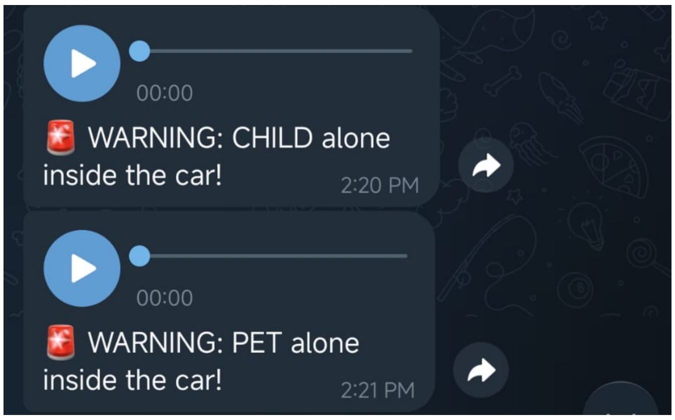
The system transmits real-time alerts to a designated caregiver via the Telegram Bot API. These notifications are categorized into two distinct modalities:

- **Visual alert** : These consist of a text-based alert accompanied by a captured frame. The image includes a bounding box for spatial localization of the detected occupant in the image.
- **Sound alert** : These comprise a text message and the specific audio recording that triggered the classification event.



🚨 WARNING: PET alone inside the car!
2:19 PM

Figure 10: Visual alert example


00:00

🚨 WARNING: CHILD alone inside the car! 2:20 PM

00:00

🚨 WARNING: PET alone inside the car! 2:21 PM

Figure 11: Sound alerts examples

The Telegram API used to send images and audios are respectively :

- https://api.telegram.org/bot{BOT_TOKEN}/sendPhoto
- https://api.telegram.org/bot{BOT_TOKEN}/sendVoice

5.3.6 Decision Logic

The decision-level fusion architecture for detecting children and pets within a vehicle cabin is depicted in Figure 12. This mechanism employs a late-fusion strategy, synthesizing independent inferences from the visual and acoustic classification pipelines to produce a unified detection result. By integrating these multimodal predictions, the system enhances detection robustness and reduces the likelihood of false negatives. The confidence thresholds are set at 0.5 for Images and 0.6 for Sound. The fusion process operates (Figure 12) follows:

- The visual module classifies frames as adult, child, pet, or empty.
- The audio module classifies sounds as adult sound, child sound, pet sound, or noise/silence.
- Detection of an adult presence results in a safe state.
- If a child or pet is detected by either modality (at this point the absence of an adult is confirmed), the system dispatch an alert.

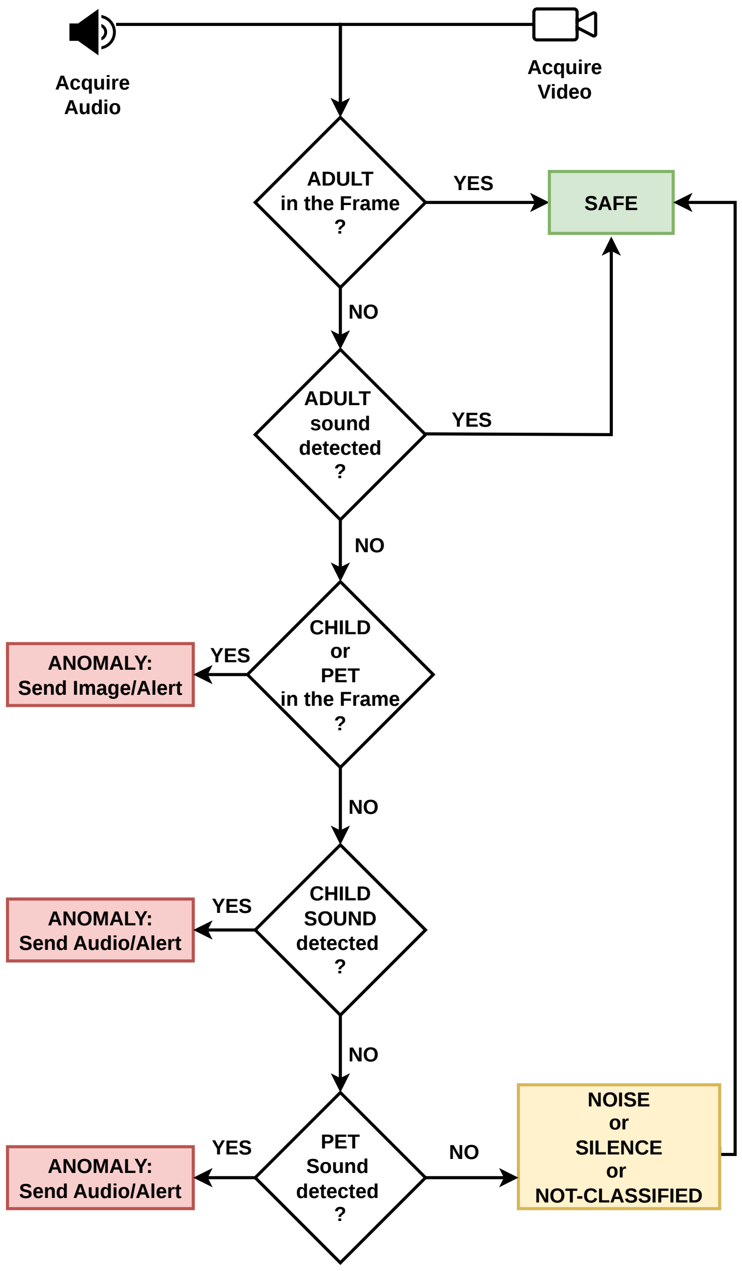


Figure 12: Decision-Level Fusion Model for Child/Pet Presence Detection

5.4 Performances and optimizations

The temporal efficiency of the proposed system was evaluated using the **Mean Total Cycle Time** ($\bar{T}_{cyc}$), defined as the average duration required to complete a single processing iteration. This cycle includes data acquisition, pre-processing, inference of both models, inference fusion and decision. To further characterize system stability, we define **Jitter** (σ) as the temporal variation between successive iterations, and **Throughput** (η) as the frequency of completed decisions (System outputs) per unit of time, expressed as:

$$\eta = \frac{1}{\bar{T}_{cyc}} \tag{1}$$

The performances taken into consideration for the system are summarized in Table 8.

Metric	**Symbol**	**Description**
Mean Cycle Time	$\bar{T}_{cyc}$	Average time elapsed from data acquisition to system output.
Jitter	σ	Standard deviation of T_{cyc}, indicating timing consistency.
Throughput	τ	Frequency of decision-making (decisions per second and decision per minute).
CPU Load	CPU_{load}	Represents the computational demand of the system's processes relative to the Raspberry Pi's processing capacity.
RAM Usage	Mem_{usage}	Indicates the RAM memory usage of system's processes.
CPU Temperature	$Temp_{cpu}$	Monitors thermal stability to ensure the system avoids frequency throttling during extended operation.

Table 8: Definitions of System Performance Metrics

The preliminary version (**V1**) of the prototype followed a synchronous, sequential execution flow. As illustrated in the control loop below, all functional modules detailed in Section §3 were executed within a single process. This sequential pipeline ensured deterministic execution but enforced a linear dependency between data acquisition and inference:

1. **Visual Acquisition & Pre-processing** (step 1): Image capture ("Eye" module).
2. **Visual Inference** (step 2): Classification of the processed frame (Image classification Module)

3. **Acoustic Acquisition** (step 3): Audio recording ("Ear" module).
4. **Acoustic Inference** (step 4): Classification of the captured audio segment (Audio classification Module)
5. **State Evaluation & Transmission** (step 5): Determination of the safety state ("Brain" module) and, if required, alert dispatch ("Communications" module).
6. **Cycle Iteration** (step 6): Return to step 1.

Resource consumption remained minimal, primarily due to the temporal requirements of the Audio Classification Module. The model necessitates 5-second audio segments for inference; consequently, the combined duration of recording and processing resulted in a $\bar{T}_{cyc}$ of 6 seconds (one output every 6 seconds). While this introduced a latency bottleneck, it significantly reduced the computational overhead on the Raspberry Pi, ensuring stable operation within the hardware's limited power and thermal envelopes. Considering the baseline performance of the Version 1, the system architecture was optimized to achieve a higher system throughput. We transitioned from a synchronous, sequential pipeline to a multiprocess execution model leveraging the Python `Multiprocessing` library. To ensure temporal alignment between different data streams, an Inter-Process Communication (IPC) `Barrier` was implemented. The system's functional modules were decoupled and deployed across three independent processes

- **"Brain" Process** (Brain module + Communications module): serves as the system's primary orchestrator, responsible for managing the life cycle of the subordinate processing units. Upon initialization, it instantiates the Eye and Ear processes and establishes the IPC Barrier mechanism to ensure strict temporal synchronization. Following this setup, the Brain Process enters a continuous execution loop to manage real-time monitoring. Within each iteration, the process issues the synchronization trigger and performs a blocking wait for the concurrent output packets from the visual and acoustic modules. Upon receiving the visual metadata and acoustic inferences, the process executes the decision-level fusion logic to determine the presence of vulnerable occupants within the cabin. If safety criteria are violated, the process invokes the Communication Module to dispatch the appropriate type of alert via the Telegram API before immediately cycling again.
- **"Eye" Process** (Eye Module + Image classification module) : Upon the one-time instantiation of the TFLite-based YOLO interpreter, the process enters its main execution loop. At the beginning of each iteration, the subsystem performs a blocking wait for a synchronization signal from the Brain

Process. Once this trigger is received, the pipeline proceeds to perform image acquisition, pre-processing, and inference. In the event of a positive detection, the process generates an annotated frame and transmits the resulting inference metadata to the decision logic via the IPC mechanism before cycling back to the synchronization point for the next iteration.

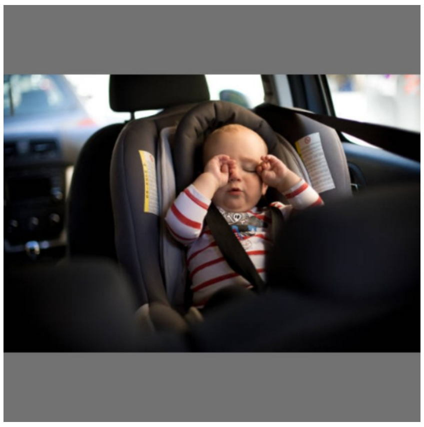
- **"Ear" Process** (Ear module + Audio Classification module) : Following the one-time instantiation of the TFLite interpreter, the process enters its primary execution loop, where it immediately performs a blocking wait for a synchronization signal from the Brain Process. Upon receiving the trigger, the subsystem initiates the acoustic acquisition phase to capture raw audio samples, which are then preprocessed . These segments are passed through the inference engine to classify specific acoustic events. Finally, the resulting inference data and the corresponding audio clip are transmitted to the Brain Process via the IPC layer before the subsystem cycles back to the initial synchronization point for the next iteration.

Switching to a parallel approach slightly improved the system's performance, bringing the $\bar{T}_{cyc}$. To push the performance even further, we looked for ways to shorten the time spent recording audio. By optimizing our audio dataset and the YAMNet+FFN architecture, we found that the model could maintain its accuracy using only 3 second audio chunks instead of 5. This update allowed the system to reach a $\bar{T}_{cyc}$ of 3.32 seconds. This faster cycle is a big advantage for safety, as it allows the system to detect and report the presence of a child or pet in nearly half the time compared to our first prototype. In Table 9 are shown the performances of the final version (**V2**).

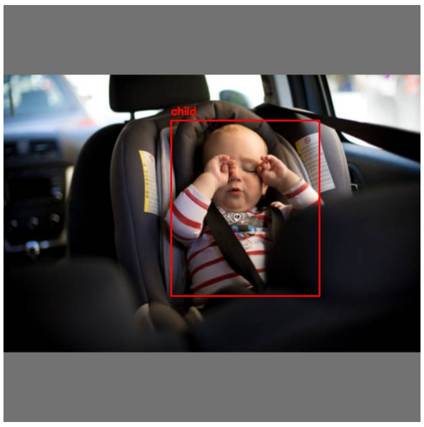
Metric (CI 99%)	**V1 (Sequential)**	**V2 (Parallel)**
$\bar{T}_{cyc}$ [s]	5.9875 ± 0.0084	3.3259 ± 0.003
σ [s]	0.037	0.199
τ [decisions/s]	0.167 ± 0.0002	0.3007 ± 0.0003
τ [decisions/min]	10.02 ± 0.01	18.04 ± 0.002
CPU_{load} [%]	16 ± 6.77	24.31 ± 4.53
Mem_{usage} [Mb]	379.75 ± 0.3294	396.8125 ± 2.5408
$Temp_{cpu}$ [°C]	55 ± 0.2505	57.7647 ± 1.7267

Table 9: Performance Metrics by Prototype Version

Preliminary field tests revealed inconsistent detection accuracy for diverse occupant categories when employing the YOLO26 Nano architecture at a 480×480 resolution. Accounting for the 3 second audio acquisition interval and the inference overhead detailed in Table 6, the 640×640 resolution was implemented. This adjustment yielded higher reliability in the final prototype without increasing the the system inference period.

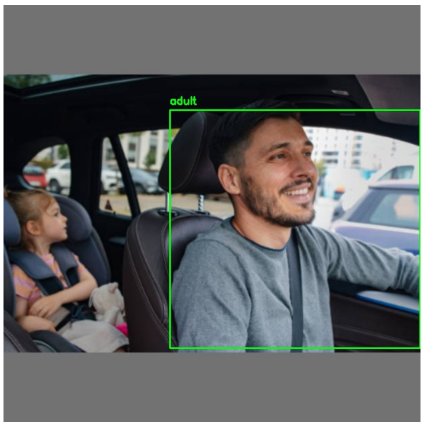


(a) Missing detections using 480x480 resolution

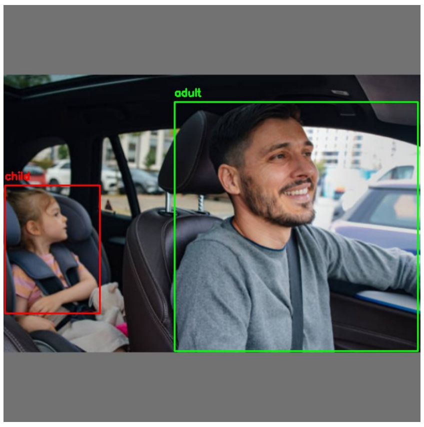


(b) Detections using 640x640 resolution

Figure 13: Differences in detections (1)



(a) Missing detections using 480x480 resolution



(b) Detections using 640x640 resolution

Figure 14: Differences in detection (2)

6 Conclusion

This project successfully developed and validated a multimodal Child and Pet Presence Detection (CPPD) prototype aimed at mitigating vehicular heatstroke caused by "Forgotten Baby Syndrome." By integrating a YOLOv26 visual model with a custom Feed-Forward Network (FFN) audio classifier—utilizing YAMNet for feature extraction on a Raspberry Pi 3B+ the proposed architecture addresses the inherent vulnerabilities of unimodal camera solutions, such as visual occlusions, blind spots, and low-light conditions.
Experimental results demonstrate that the transition from a sequential to a parallel processing architecture was critical for real-time performance. This optimization reduced the system's detection latency from approximately 6 seconds to 3.32 seconds, significantly enhancing the system's responsiveness. The prototype effectively implements decision-level fusion to distinguish between safe scenarios (adult presence) and critical risks (unattended children or pets), triggering immediate alerts via the Telegram API.
Ultimately, this study confirms that a low-cost, edge-computing solution can achieve reasonable detection accuracy and fast response times required for automotive safety standards, providing a viable technological safeguard against preventable fatalities.

7 Code availability

The source code for the analysis and models in this study is openly available in the **Guardian** repository at **https://github.com/CarloPace/Guardian**. The datasets are not publicly available due to legal and copyright restrictions.

References

[1] C.-H. Choi, S. Hong, E. J. Jeong, and J. Han, “Infrared thermal imaging for embedded child presence detection system: Feasibility and performance evaluation,” in *Proceedings of the IEEE International Conference on Consumer Electronics Asia (ICCE-Asia)*, Danang, Vietnam, 2024.

[2] J. Gillner, R. Taylor, and K. Thelen, “Radar-based breathing sensor for child presence detection in automotive applications,” in *2024 IEEE 67th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2024, pp. 104–107.

[3] M. A. H. M. Nazri, S. A. B. C. Abdullah, A. Ahmad, F. H. K. Zaman, N. K. Mun, and M. F. M. Salleh, “In-vehicle child presence detection using 60ghz radar,” in *2025 IEEE 8th International Conference on Electrical, Electronics, and System Engineering (ICEESE)*. IEEE, 2025, pp. 42–47.

[4] Y. Su, F. Zhang, K. Niu, T. Wang, B. Jin, Z. Wang, Y. Jiang, D. Zhang, L. Qiu, and J. Xiong, “Embracing distributed acoustic sensing in car cabin for children presence detection,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 8, no. 1, Mar. 2024.

[5] J. Cao, B. Peng, M. Gao, H. Hao, X. Li, and H. Mou, “Object detection based on cnn and vision-transformer: A survey,” *IET Comput. Vis.*, vol. 19, p. e70028, 2025.

[6] M. L. Ali and Z. Zhang, “The yolo framework: A comprehensive review,” *Computers*, vol. 13, no. 12, p. 336, 2024. [Online]. Available: https://doi.org/10.3390/computers13120336

[7] R. Sapkota, R. H. Cheppally, A. Sharda, and M. Karkee, “Yolo26: Key architectural enhancements and performance benchmarking for real-time object detection,” *arXiv preprint arXiv:2509.25164*, 2026.

[8] T. Yerulan and D. Zhasdauren, “Comparative study of lightweight object detection models for mobile pedestrian detection,” *Universum*, vol. 10, no. 5 (134), pp. 4–8, 2025.

[9] H. Van Thanh *et al.*, “An efficient object detection model based on nanodet,” in *2025 International Workshop on Intelligent Systems (IWIS)*. IEEE, 2025.

[10] M. K. Gourisaria *et al.*, “Comparative analysis of audio classification with mfcc and stft features using machine learning techniques,” *Discover Internet of Things*, vol. 4, no. 1, p. 1, 2024.

[11] N. H. Valliappan, S. D. Pande, and S. R. Vinta, "Enhancing gun detection with transfer learning and yamnet audio classification," *IEEE Access*, vol. 12, pp. 58 940–58 949, 2024.

[12] L. Liu *et al.*, "Yamnet-based transfer learning for compact noise classification in urban and wireless systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2025, no. 1, p. 74, 2025.

[13] A. S. Podda *et al.*, "Cargram: Cnn-based accident recognition from road sounds through intensity-projected spectrogram analysis," *Digital Signal Processing*, vol. 147, p. 104431, 2024.

[14] H. Sakaki *et al.*, "Comparative evaluation of cnn, vgg16, conformer, and yamnet models for skateboard riding sound detection," in *2025 International Symposium on Multimedia and Communications Technology (ISMAC)*. IEEE, 2025.

[15] Raspberry Pi Foundation, *Raspberry Pi 3 Model B+ Product Brief*, Raspberry Pi Foundation, 2018, accessed: 2026-02-21. [Online]. Available: https://datasheets.raspberrypi.com/rpi3/raspberry-pi-3-b-plus-product-brief.pdf

[16] Western Digital Technologies, Inc. / SanDisk, *SanDisk Ultra microSD UHS-I Card Datasheet (Up to 150 MB/s)*, Western Digital Technologies / SanDisk, 2023, accessed: 2026-02-21. [Online]. Available: https://documents.sandisk.com/content/dam/asset-library/en_us/assets/public/sandisk/product/memory-cards/ultra-uhs-i-microsd/data-sheet-ultra-uhs-i-microsd-150mbps.pdf

[17] Raspberry Pi Foundation, *Raspberry Pi Camera Module v2 Datasheet / Product Brief*, Raspberry Pi Foundation, 2016, accessed: 2026-02-21. [Online]. Available: https://www.raspberrypi.com/documentation/accessories/camera.html