Article

Low-Cost Multimodal Pipeline for Curbside Vehicle CO_2 Emission Estimation

Rytis Paškauskas[1], Serafina Di Gioia[2], Abdullah Aloud [3], Joud Aldhuwaihi[4]

[1] Abdus Salam International Centre for Theoretical Physics (ICTP), Trieste, Italy; rytis.paskauskas@ictp.it
[2] Scuola Internazionale Superiore di Studi Avanzati (SISSA), Trieste, Italy; sdigioia@sissa.it
[3] King Saud University , Riyadh , Saudi Arabia abdullah.nsq1@gmail.com
[4] Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia; joudah2003@gmail.com

Abstract

Accurately estimating vehicle-specific CO_2 emissions using passive, non-invasive sensing remains an open challenge in computer vision and Deep learning. This paper presents a scalable, low-cost pipeline that combines visual and audio data collected from the roadside to estimate emissions without onboard sensors. A Raspberry Pi 4B equipped with a USB camera, external microphone, sound-triggered recording logic, and LED indicator captures short, synchronized video–audio clips only when a passing vehicle is acoustically detected. Vehicles are detected and cropped using YOLOv8, then classified into one of 4,481 modern car models via an EfficientNetV2-M trained on a curated dataset combining VMMRdb and manually collected images, excluding pre-2000 vehicles. Predicted models are mapped to certified CO_2 ratings from the UK Vehicle Certification Agency's Euro 6 database and the European Environment Agency's 2024 provisional CO_2 emissions dataset, with estimated vehicle speed from frame analysis used as a secondary adjustment factor. Based on prior research in acoustic engine classification, we explored adding a fuel-type inference step (diesel vs. gasoline) to improve estimates for models with multiple variants; however, high outdoor noise prevented its inclusion in the final results. Trained on the Leonardo HPC platform and evaluated on real-world data from Italy, the system achieved a ~90.7% weighted F1 score, demonstrating the potential of edge-deployed, multimodal sensing for urban-scale emissions monitoring.

Keywords: vehicle CO_2 emissions; vehicle classification; computer vision; TinyML; YOLO; EfficientNet; multimodal sensing; edge AI

Academic Editor: Firstname Lastname

Received: date
Revised: date
Accepted: date
Published: date

Citation: To be added by editorial staff during production.

Copyright: © 2025 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The growing global commitment to mitigating climate change has intensified the need for accurate monitoring and regulation of greenhouse gas emissions, particularly carbon dioxide (CO_2) from road transport. Private vehicles are a major source of these emissions, yet existing measurement techniques—such as onboard diagnostics (OBD) and laboratory-based testing—are impractical for large-scale urban deployment. This challenge has motivated research into alternative approaches that use computer vision and

machine learning to infer vehicle characteristics, including type, make, and emissions, from visual and audio data [1,2].

However, to our knowledge, no prior work has directly addressed the problem of estimating a vehicle's CO_2 emissions after visually identifying its model, leaving a key gap that this study addresses. We present a lightweight, sensor-free pipeline that estimates vehicle-specific CO_2 emissions using only a camera and microphone on a Raspberry Pi. The system employs a sound-triggered recording mechanism, activating the camera when a passing vehicle is detected acoustically. This configuration—featuring an external microphone, LED indicator, and physical trigger—enables efficient edge-level data collection while minimizing unnecessary power consumption.

Fine-grained vehicle classification is achieved using an EfficientNetV2-M model trained on a curated and extended version of the Vehicle Make and Model Recognition Database (VMMRdb), reduced from over 9,000 classes by removing pre-2000 vehicles, then supplemented with 398 additional modern models scraped from the internet, resulting in a total of 4,481 classes [3]. The dataset is cross-referenced with official CO_2 emission values from both the UK Vehicle Certification Agency's Euro 6 database and the European Environment Agency's 2024 provisional CO_2 emissions dataset [4,5]. Emission estimates are derived by mapping the predicted model (and, when available, engine type) to these certified ratings, with estimated vehicle speed from video analysis incorporated as a secondary factor.

Following prior research in acoustic engine classification [6] we explored adding an audio-based fuel-type inference step (diesel vs. gasoline) to improve accuracy for models with multiple variants. However, environmental noise in street recordings prevented its use in the final results, highlighting a key direction for future work.

The system was deployed and tested on real-world street data collected in Italy, with model training conducted on Leonardo, one of Europe's most powerful high-performance computing (HPC) platforms [5]. By combining low-cost edge hardware, deep learning, and official emissions datasets, this work proposes a scalable framework for urban CO_2 monitoring, supporting future smart city and air quality management initiatives.

2. Materials and Methods

2.1 Raspberry Pi Setup

The hardware platform was built on a Raspberry Pi 4B configured for real-time, edge-level processing suitable for TinyML applications. A Logitech USB webcam with an integrated microphone was used to capture synchronized video and audio streams, while a GPIO-connected LED indicator provided visual feedback during recording.

Raspberry Pi OS was installed and configured for remote access via Secure Shell (SSH) from an Ubuntu virtual machine. Initial setup included testing all peripherals to ensure the camera–microphone unit was fully operational. A sound-triggered recording mechanism was implemented to initiate a 10-second synchronized capture whenever engine-like audio peaks exceeded a predefined threshold, reducing unnecessary storage use and power consumption.

2.2 YOLO-Based Car Detection

Captured videos were processed using the YOLO object detection framework. Initial experiments employed YOLOv5 to perform frame-by-frame inference, detect vehicles, crop bounding boxes, and save each detected vehicle as an independent image. To improve detection accuracy and reduce missed instances, the pipeline was later upgraded to YOLOv8, which provided higher precision and faster inference.

This detection stage ensured that only relevant vehicle crops were forwarded to the classification module, filtering out non-vehicle frames and reducing computational overhead in subsequent processing.

2.3 Car Dataset Curation

The classification model was trained on a combined dataset composed of the publicly available Vehicle Make and Model Recognition Database (VMMRdb) and a custom dataset collected via web scraping using SerpAPI [8]. Starting from over 9,000 classes in VMMRdb, all vehicles manufactured before 2000 were removed, and 398 modern car models were added from the custom collection, resulting in a curated set of 4,481 models and more than 180,000 images. The dataset was refined through label harmonization to ensure consistent naming aligned with official vehicle records, removal of duplicate entries, and manual inspection to discard low-quality or mislabeled images. Each model in the final dataset was then cross-referenced with the UK Vehicle Certification Agency (VCA) database to associate it with an official CO_2 emission rating.

2.4 HPC Model Training

Fine-grained vehicle classification was carried out using an EfficientNetV2-M architecture implemented in PyTorch. EfficientNetV2-M was selected for its strong balance between classification accuracy and computational efficiency, making it well-suited for large-scale, fine-grained recognition tasks with thousands of classes.

Model training was executed on the Leonardo supercomputer (CINECA, Italy), equipped with NVIDIA A100 GPUs and managed through a Slurm-based scheduling environment, enabling large-scale, high-speed computation.

The curated dataset of 4,481 vehicle models and over 180,000 images was used for training. To improve generalization and robustness, we applied extensive data augmentation (including random rotations, flips, color jittering, and scaling). Mixed-precision training was employed to accelerate computation while reducing GPU memory usage. Hyperparameters such as learning rate, batch size, and optimizer settings were tuned through multiple trial runs to maximize performance.

Final model performance reached a top-1 accuracy of 90.66%, with weighted precision, recall, and F1 scores all exceeding 90%, demonstrating strong classification capability across a large number of fine-grained classes.

2.5 Audio Processing Pipeline

Audio streams extracted from the recorded videos were processed using Librosa to generate spectrogram representations. Environmental noise reduction filters were applied to suppress background sounds, followed by blob detection to identify and segment individual vehicle pass-by events.

For each detected event, the corresponding audio segment was paired with the vehicle image obtained from the detection stage, ensuring synchronized multimodal data. The pipeline was designed to include a clustering-based classification method for distinguishing between diesel and gasoline engines, an approach inspired by prior research in acoustic engine analysis. This step aimed to refine CO_2 emission mapping for models available in multiple engine variants.

In practice, however, high levels of environmental noise—such as wind, overlapping traffic sounds, and pedestrian activity—significantly reduced clustering reliability. As a result, this method was excluded from the final CO_2 estimation stage, though it remains a promising direction for future work if cleaner audio data can be collected.

2.6 CO_2 Emission Estimation

CO_2 emissions were estimated by mapping each predicted vehicle model—along with its inferred engine type when available—to official CO_2 ratings from the UK Vehicle Certification Agency's Euro 6 dataset, supplemented with the 2024 European Environment Agency (EEA) CO_2 emissions database.

To account for operational context, vehicle speed was estimated using optical tracking of the vehicle's displacement across consecutive frames within a calibrated field of view. This calibration ensured that pixel-based movement could be accurately translated into real-world speed.

The final CO_2 estimate combined the model-specific regulatory emission rating with a speed-dependent adjustment factor, producing values that better reflect realistic on-road driving conditions rather than static catalog specifications.

2.7 Summary of the Full Pipeline

The proposed system integrates all processing stages into a sequential pipeline for vehicle CO_2 emission estimation, as illustrated in Figure 1.

Data acquisition starts with the Raspberry Pi setup, where a sound-triggered mechanism activates the camera–microphone unit to capture short, synchronized video–audio clips of passing vehicles.

The recorded videos are first processed by the YOLO-based detection module, which identifies vehicles, crops their bounding boxes, and saves each as an independent image. These cropped images are then passed to the classification module, where an EfficientNetV2-M model—trained on the combined VMMRdb and custom dataset (Section 2.3)—predicts the make and model of each vehicle. Model training was carried out on the Leonardo HPC platform (Section 2.4).

In parallel, the audio processing pipeline (Section 2.5) isolates engine sound events from the recorded clips. A clustering step for engine-type identification (diesel vs. gasoline) was planned but omitted in the final results due to excessive environmental noise.

Finally, the CO_2 estimation module (Section 2.6) maps each predicted vehicle model—and engine type, when available—to official CO_2 ratings from the Euro 6 dataset

and the 2024 EEA emissions database. Estimated vehicle speed, obtained from video-based motion tracking, is used as a secondary adjustment factor to produce more realistic on-road emission estimates.

This end-to-end pipeline demonstrates the feasibility of a low-cost, deployable framework for urban CO_2 monitoring, combining edge-based sensing, computer vision, and regulatory datasets into a unified, scalable solution.

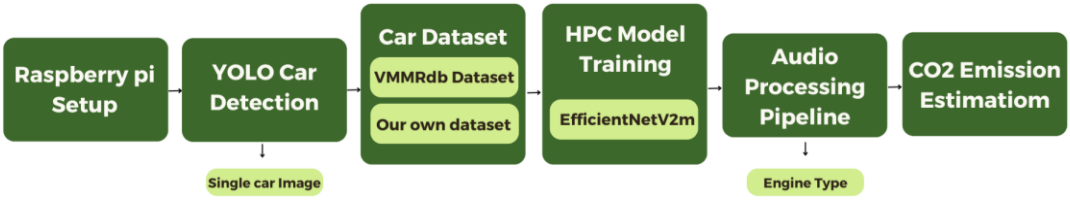


Figure 1 Full Pipeline

3. Results

3.1 Vehicle Detection and Capture

The YOLO-based module successfully detected and cropped vehicles from sound-triggered Raspberry Pi recordings (Fig. 1). Moving from YOLOv5 to YOLOv8 reduced missed detections qualitatively in street scenes and produced cleaner crops for the classifier. The crossing-line tracker generated, per car, a best-quality crop and an estimate of traversal time and speed; these were saved together with synchronized 1-second audio clips.

3.2 Fine-Grained Vehicle Classification (4481 Classes)

Using EfficientNetV2-M with a 1-channel input and the curated VMMRdb + custom dataset, we performed 5-fold cross-validation, training the model for 95 total epochs. Aggregate metrics across all folds:

Accuracy: 0.9066
Precision (weighted): 0.9051
Recall (weighted): 0.9066
F1 Score (weighted): 0.9030

Per-fold validation accuracy (best epoch recorded per fold):
Fold 1: 0.9123
Fold 2: 0.8643
Fold 3: 0.9042
Fold 4: 0.9240
Fold 5: 0.8777

Training accuracy rose from ~0.87 to ~0.91 over the course of training, indicating healthy convergence without evidence of overfitting collapse.

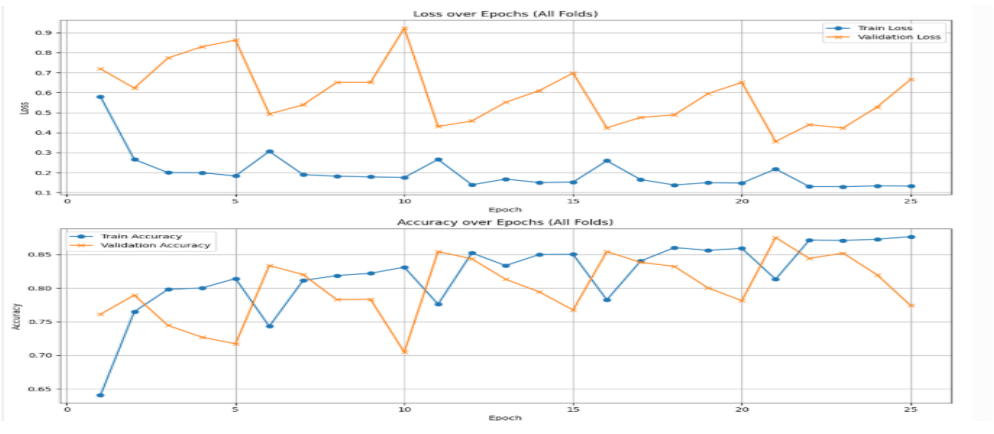


Figure 2 illustrates the training and validation loss (top) and accuracy (bottom) trends across epochs, aggregated over all folds. The plots confirm steady improvement in accuracy and a decreasing loss pattern for both training and validation, with no signs of severe overfitting.

3.3 Audio Processing

The pipeline extracted mono audio, segmented 1-second windows around vehicle pass-bys, and paired them with the corresponding image crops. A clustering step to separate gasoline vs. diesel engines (MFCC-based) was prototyped but not used in the final estimates due to unreliable separation under outdoor noise (wind, ambient traffic, conversations).

3.4 CO_2 Emission Lookup and Outputs

For each detected car, the predicted model index was mapped to a short label and then linked to a VCA Euro 6 table to obtain a per-model CO_2 estimate (g/km). When available, the pipeline recorded estimated vehicle speed from the crossing duration; this value is retained as a secondary modifier/feature for future work. Final per-car outputs are stored in JSON alongside crops and audio segments.

4. Discussion

Our results show that high-accuracy, fine-grained vehicle recognition can be achieved at low cost using compact, edge-friendly architectures. Achieving a ~90.7% weighted F1 score across 4,481 modern models after 95 training epochs demonstrates that a single-channel EfficientNetV2-M, trained on a carefully curated dataset, can deliver robust make–model predictions under real-world curbside conditions. The deliberate label harmonization and removal of pre-2000 vehicles reduced inter-class ambiguity and improved recognition stability.

Detector choice as a downstream enabler. The move from YOLOv5 to YOLOv8 improved both crop quality and recall, directly benefiting classification consistency. In our setup, adjusting detection intervals and minimum bounding-box sizes helped balance precision and recall while keeping computational demands manageable — a critical factor for embedded or large-scale deployments.

Audio is valuable but noise-sensitive. Although literature supports acoustic separation of fuel types, our street deployment highlighted the fragility of unsupervised clustering under environmental noise. Reliable fuel-type inference may require:

(i) stronger denoising,

(ii) supervised training with labeled engine audio, or

(iii) multi-second context models (e.g., CNN-TDNN architectures) rather than short MFCC windows.

From class to emissions. Mapping predicted models to certified CO_2 values creates a practical bridge between computer vision and regulatory inventories. Where models have multiple trims/engines, disambiguation (via audio fuel-type, displacement cues, or license-plate-to-VIN services where legally permitted) would reduce variance. Speed estimation was retained but not yet integrated into a complete emissions-vs-velocity model; adding this would move from catalog values toward operating-condition estimates.

Failure modes observed:

Occlusions or partial crops causing confusions among visually similar variants.

Long-range shots reducing fine-detail needed for variant separation.

Nighttime and strong backlight conditions degrading both detection and classification.

Audio segments contaminated by wind and multi-source noise collapsing diesel/gasoline cluster separability.

Current limitations.

Our cross-validation was performed on a curated dataset, and broader generalization to new cities, camera placements, and environmental conditions will require domain-shift evaluation. Emission outputs currently reflect catalog ratings per model rather than instantaneous emission rates. Finally, in this release, the audio module is extraction-only; fuel-type inference is deferred to future work.

5. Conclusion

This work demonstrates that low-cost, passively deployed sensing can meaningfully contribute to urban-scale CO_2 monitoring without the need for intrusive, vehicle-mounted equipment. By combining curbside imagery and audio with fine-grained machine learning models, we showed that it is possible to reliably identify 4,481 modern car models and link them to certified Euro 6 CO_2 ratings, achieving ~90.7% accuracy and 90.3% weighted F1 through an EfficientNetV2-M classifier trained for 95 epochs with 5-fold cross-validation.

The Raspberry Pi–based rig, paired with YOLOv8 detection and crossing-based speed estimation, generated structured, per-car records that could feed directly into air quality dashboards, city traffic models, and policy evaluation tools. In real-world street tests, the system produced actionable emissions estimates while operating on inexpensive, edge-level hardware — opening possibilities for dense, distributed deployments across urban areas.

Although our initial attempt at unsupervised fuel-type clustering proved unreliable under noisy outdoor conditions, the modular design of the pipeline enables clear paths for enhancement. Future developments include:

(i) integrating supervised audio models for fuel-type and engine displacement inference,

(ii) incorporating speed/load effects via physics-informed or learned emission curves, and

(iii) evaluating domain adaptation methods to ensure generalization across cities, climates, and camera setups.

By bridging computer vision, acoustics, and regulatory emissions inventories, this framework moves a step closer to scalable, data-driven environmental intelligence. In the long term, such systems could empower municipalities, researchers, and policymakers to measure, analyze, and ultimately reduce road-transport emissions at the granularity needed for effective climate action..

References

1. S. B. Neamah and A. A. Karim, “Vehicle Classification and Counting for Traffic Analysis based on Single-stage YOLOv8 Model,” Iraqi Journal of Computer Communication Control and System Engineering, vol. 24, no. 2, pp. 41–53, Jun. 2024, doi: https://doi.org/10.33103/uot.ijccce.24.2.4.
2. Md. Asifur Rahman Khan, R. Talukder, Md. Anwar Hossen, and N. Jahan, “Car make and model recognition using convolutional neural network: fine-tune AlexNet architecture,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 33, no. 1, p. 370, Jan. 2024, doi: https://doi.org/10.11591/ijeecs.v33.i1.pp370-379.Author 1, A.; Author 2, B. *Book Title*, 3rd ed.; Publisher: Publisher Location, Country, 2008; pp. 154–196.
3. Faezetta, “GitHub - faezetta/VMMRdb: Vehicle Make and Model Recognition Dataset (VMMRdb),” GitHub. https://github.com/faezetta/VMMRdb
4. Vehicle Certification Agency, “Select a search : Directgov - Car fuel data, CO2 and vehicle tax tools,” carfueldata.vehicle-certification-agency.gov.uk. https://carfueldata.vehicle-certification-agency.gov.uk/
5. European Environment Agency, “Monitoring of CO_2 emissions from passenger cars,” 2024. [Online]. Available: https://co2cars.apps.eea.europa.eu/.
6. Mateusz Materlak and Ewelina Majda-Zdancewicz, “Classification of Engine Type of Vehicle Based on Audio Signal as a Source of Identification,” Electronics, vol. 12, no. 9, pp. 2012–2012, Apr. 2023, doi: https://doi.org/10.3390/electronics12092012.
7. CINECA Leonardo Supercomputer, “Home | Leonardo Pre-exascale Supercomputer,” Leonardo CINECA , Apr. 21, 2022. https://leonardo-supercomputer.cineca.eu/
8. “SerpAPI: Google Search API,” SerpApi. https://serpapi.com/