SoundDetectClass – Sound Events Inference for Hospitals

Joel Paulo[a,b,c], André Carrasqueira[a]

***Abstract*— Noise in hospital intensive care units is a serious, ongoing problem that affects patients. Noise levels (LAeq) in a hospital environment should not exceed 35/40 dBA during the day and 30 dBA at night (Berglund et al., 1999; WHO, 2018). However, quantifying this type of discomfort based solely on sound levels does not fully reflect reality, as it is not assessed objectively. Indeed, certain types of noise produce low average sound levels over extended periods of the day or night, yet they cause significant discomfort to patients. This project consists of developing an intelligent system, called SoundDetectClass, for detecting and classifying sound events in hospital environments, with a primary focus on Intensive Care Units (ICUs). The work enables non-intrusive sound monitoring, automatically classifying sounds such as voices, screams, alarms, music, or equipment noise, and correlating them with continuous, real-time sound levels. This solution implements Artificial Intelligence models trained with the AudioSet dataset, such as YAMNet, PANNs, and urban8kECAPA, and has been quantitatively evaluated based on the dataset HospitaldB, built with sound events captured in the hospital environment, and real-time testing. The HospitaldB dataset consist of more than 50 annotated sounds belonging to 10 families, often present in hospital environments: speech, screams, alarm, telephone, music, whistle, wheels, snore, waterfall and impact. For each family, we analyzed the distribution of scores assigned by the different predefined classes of the classifiers. The system performs inference efficiently on devices such as the Raspberry Pi and stores results with confidence indicators. A mapping was also created between the 527 AudioSet classes and 10 sound families relevant to the hospital context, allowing for significant abstraction in human data interpretation. The results obtained from this work allow us to confirm that generalist audio classification models, such as YAMNet, can be successfully adapted to specific domains simply by changing the indexes and names of the families. The HospitaldB dataset, the mapping of classes to families, and the introduction of relevance weights are central elements in achieving this goal. Furthermore, the project revealed the importance of iterative testing and validation at multiple levels: from testing with long and complex sound files to inference analysis using short-duration blocks. This approach allowed us to determine that, although models such as PANNs perform robustly in noisy environments, YAMNet was better suited to the type of discrete, well-defined events found in hospital settings. The successful execution of the classifier on a Raspberry Pi 3 model B+ with 1GB, minicomputer, with multi-processing integration and MQTT transmission to a dashboard on the cloud, confirms the viability of the solution for implementation in real-world scenarios. The modularity of the code, with a clear separation between capture, classification, and communication, facilitates adaptation to other environments, such as schools, factories, or public spaces.**

[a]ISEL - Instituto Superior de Engenharia de Lisboa, Portugal
[b]LAA-Audio and Acoustics Laboratory, Instituto Superior de Engenharia de Lisboa, Portugal
[c]NOVA LINCS, NOVA School of Science and Technology, Monte da Caparica, Portugal

***Keywords*— Noise in Hospitals; continuous sound monitoring; machine learning; sound event detection and classification; IoT, RaspberryPi.**

I. Introduction

In recent years, the automatic detection and classification of sound events has gained prominence, particularly in areas such as intelligent surveillance, home automation, environmental analysis, urban monitoring, among others. These systems are based on machine learning technologies, audio processing and artificial intelligence to recognize sound patterns and assign them to specific contexts such as music, speech, alarms, nature sounds or impacts.

The project developed is part of this technological area, aiming at the creation of a real-time sound event detection system, with a special focus on hospital environments, the SoundMeterHosp, SMH [1]. This system allows the identification of significant events (speech, music, alarms, etc.) and their cataloging in real time, enabling context analysis, dynamic reports, and integration with external systems.

In fact, the SoundMeterHosp, SMH, project proposes the development of an integrated solution for sound monitoring in hospital environments, focusing on data privacy, modularity, and ease of use. The architecture integrates sound analysis stations, including a sound traffic light approach, to show through colored lights 3 ranges of variation of sound levels, and an online platform for storing, analyzing and visualizing results. This platform uses an IoT architecture based on the MQTT communications protocol; processing with Node-RED; data storage in InfluxDB; and visualization in interactive dashboards via Grafana, accessible through a web application. Remote reconfiguration and display functionality allow the solution to be adapted to different clinical scenarios.

The system was designed to be used in hospitals to inform nurses of events that occur when patients are physically alone. With this system, dripping water, knocks, alarms can be stopped before the patient has to get up.

II. Overview

A. Sound Event Detection

Sound Event Detection (SED) is an active research field that intersects with areas such as pattern recognition, signal analysis, deep neural networks, and semantic classification. There are different approaches to SED, depending on the objective (real-time vs. archive), type of environments (indoor, outdoor, urban), and granularity of the events to be identified (broad sounds vs. specific sounds).

Examples of identifying types of sound events, such as noise

generated by vehicles on various types of road surfaces [2], impulsive noise [3], detection of knock situations [4].

There are a number of projects and systems similar to our approach in urban environments which uses low-cost sensors to monitor urban noise (smart cities).

The SONYC (Sounds of New York City) aims to create technological solutions for: (1) the systematic, constant monitoring of noise pollution at city scale; (2) the accurate description of acoustic environments in terms of its composing sources; (3) broadening citizen participation in noise reporting and mitigation; and (4) enabling city agencies to take effective, information-driven action for noise mitigation [5]. Another examples, developed in 2015 and 2020 by the author, refers to the FI-Sonic Project which refers to the processing and analysis of sound captured by a network of multichannel microphones distributed throughout the city. By using advanced and innovative 3D audio capturing and processing technologies and machine learning techniques, FI-Sonic assesses the sound field and extracts the relevant information about the sound events occurring in the city such as excessive sound levels, accidents, gun shots, distress situations, among other city sounds [6, 7].

Some of the best-known frameworks for this task include:

YAMNet (Yet Another Mobile Network): classification model based on training with datasets (Audioset). Classifies audio segments by assigning a score to each of the categories for which it was trained [8, 9].

VGGish: another Google architecture that focuses on more generic, but less specific, audio features [10].

OpenL3: audio features trained with deep networks in multimodal tasks (visual and sound). Similar to VGGish but has a more computationally intensive approach for simultaneous detection in different modes [11].

B. *Dataset Audioset*

AudioSet is an extensive database maintained by Google, containing more than 2 million audio samples with semantic annotations categorized into 527 classes. Each file is associated with one or more classes (for example, "laughing," "car horn," "dog bark"), making it ideal for training and testing general audio classification models [12, 13].

Despite the richness of the dataset, its complexity requires adaptations. It is a user-created database with little moderation, however, no categorization is wrong. The more general classes ("speech", "music") have many more samples added by users.

Less popular and more specific classes ("man speaking", "Jazz") often end up being ignored. This is important because models trained with Audioset will be better at classifying classes with more examples in the dataset than those with fewer.

class_labels_indices.csv

```
index,mid,display_name
0,/m/09x0r,"Speech"
1,/m/05zppz,"Male speech, man speaking"
2,/m/02zsn,"Female speech, woman speaking"
3,/m/0ytgt,"Child speech, kid speaking"
4,/m/01h8n0,"Conversation"
5,/m/02qldy,"Narration, monologue"
6,/m/0261r1,"Babbling"
7,/m/0brhx,"Speech synthesizer"
8,/m/07p6fty,"Shout"
9,/m/07q4ntr,"Bellow"
10,/m/07rwj3x,"Whoop"
11,/m/07sr1lc,"Yell"
12,/m/04gy_2,"Battle cry"
13,/t/dd00135,"Children shouting"
14,/m/03qc9zr,"Screaming"
15,/m/02rtxlg,"Whispering"
16,/m/01j3sz,"Laughter"
17,/t/dd00001,"Baby laughter"
18,/m/07r660,"Giggle"
```

Fig. 1. classes labels indices.csv: The first 35 classes in AudioSet.

C. *Requirements*

The developed system meets the following functional requirements:

- Audio capture: The system must capture ambient sound using microphones connected to an acquisition interface.
- Sound classification: It must identify the main sound events using an AI model.
- Semantic grouping: It must reorganize the detected events into groups to facilitate their interpretation. These groups must summarize the possible sound events in a hospital environment.
- Database: To effectively map families, a set of sounds from each family must be gathered to define the classes that constitute each family. These sounds must be different from those used in training the model.
- Configuration: This project must be delivered in a structured way, in order to encourage changes to the values used in the model.
- Local visualization: There must be a possibility of local visualization of the captured values.
- Real-time performance: Audio processing must be performed with minimal delay (less than 1 second).
- Modularity: The system must be segmented into parallel processes to facilitate maintenance and scalability.
- Robustness: In case of failure in one module, the others must remain operational.
- Portability: The system must run on IoT microcomputers with other integrated components (e.g., Raspberry Pi).
- Documentation: The code must be commented and accompanied by technical and user documentation.

III. Selection of the Model

The approach followed was based on an iterative experimental principle: it was necessary to determine not only

which was the best audio classification model, but also how to represent the classification results in an interpretable and semantic way.

The process generally consisted of the following steps:

- Classifier Selection: It started with YAMNet, as it is the lightest reference model with direct integration in Python. Subsequently, the alternative models urban8k-ECAPA (a model optimized for only 8 urban classes) and PANNs-inference (a difficult model, but with high robustness in complex environments) were explored [8];
- Tests with a Custom Database: The HospitaldB database was created, a database with real hospital sounds, segmented by thematic families, (partially shown in Figure 2);
- Construction of a Semantic Structure: To interpret the raw scores of the classifiers, a family structure was developed. .csv files were defined with the mapping of classes to families and, additionally, a weighting system that reflects the relevance of each class within its family;
- Filtering of Irrelevant Classes: The need to ignore classes such as "Silence" was identified, resorting to the implementation of a filter that does just that;
- Validation with Small Windows: Tests were performed with 1-second audio blocks to simulate real-time execution;
- Integration with Raspberry Pi: The selected classifier (YAMNet) was isolated in a multiprocessor and validated on Raspberry Pi;

The performance of these classifiers was made into the Python environment. For each classifier, different obstacles were encountered:

- YAMNet: The model is available as a SavedModel in Tensorflow [18].
  It required configuring the environment with Tensorflow 2.x, and using audio samples with a compatible shape (16 kHz, mono).
- PANNs-inference: Although more accurate with long and complex sounds, its use implied dependencies such as torch and torchlibrosa [19]. It was enough to install the Python panns-inference library; however, for operation, it was necessary to install a 300 MB .pth file in the directory C:/User/panns-data named Cnn14-mAP=0.431.pth [14];
- urban8k-ECAPA: Lightweight but highly limited model: only correctly classifies the 8 classes for which it was trained. Its implementation required the installation of the Python Speechbrain library. It is a model not installed locally. There was no problem with its installation [15, 16].

Families				
speech	Speech	Male speech, man speaking	Female speech, woman speaking	...
screams	Shout	Bellow	Whoop	...
alarm	Car alarm	Reversing beeps	Police car (siren)	...
telephone	Telephone	Telephone bell ringing	Telephone dialing, DTMF	...
music	Singing	Choir	Yodeling	...
whistle	Whistling	Train whistle	Whistle	...
wheels	Carnatic music	Scary music	Vehicle	...
snore	Groan	Grunt	Breathing	...
waterfall	Water	Rain	Raindrop	...
impulsive	Run	Shuffle	Walk, footsteps	...

Fig. 2. Families with classes assigned from the AudioSet.

All models were tested with .wav files of real sounds to ensure they worked in a realistic environment.

In order to test the accuracy of the pre-trained models a compilation of various types of sounds from different categories, such as, scream, wheels, impulsive, alarm, snore and music was used in the tests (Master.wav). Figure 3 shows the 10 classes with the highest scores for each classifier (Top10) for each classifier considered in this study.

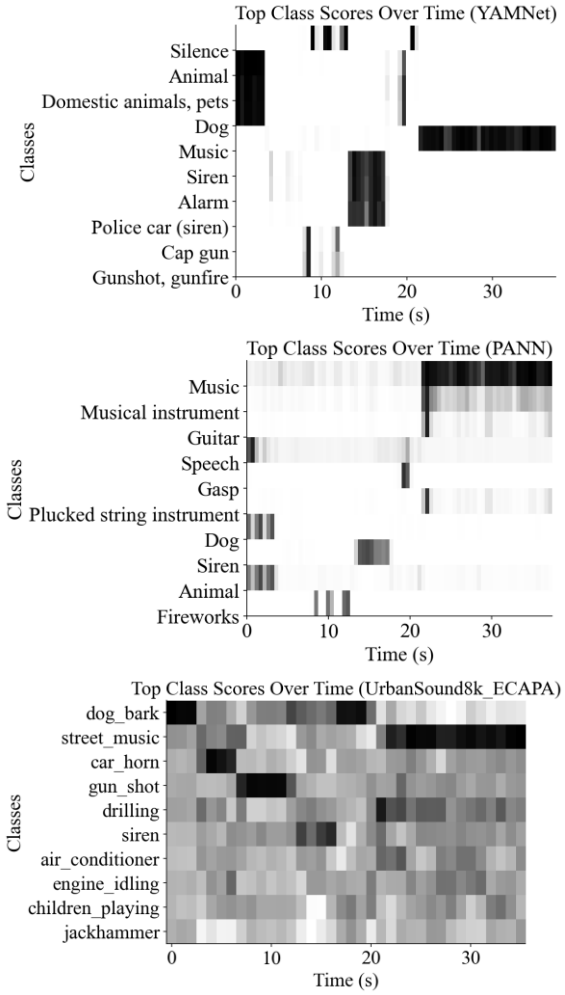


Fig. 3. Top10 classes over time for YAMNet, PANN and UrbanSound8k_ECAPA.

The UrbanSound8k_ECAPA classifier is only built for the ten classes shown in Figure 3 (lower image) [9]. However, some classes are not relevant to normal activities in a hospital environment. Therefore, it will not be considered in the following study.

Table I shows the performance of the tested classifiers, YAMNet and PANN (CNN14).

TABLE I: CLASSIFIERS PERFORMANCE

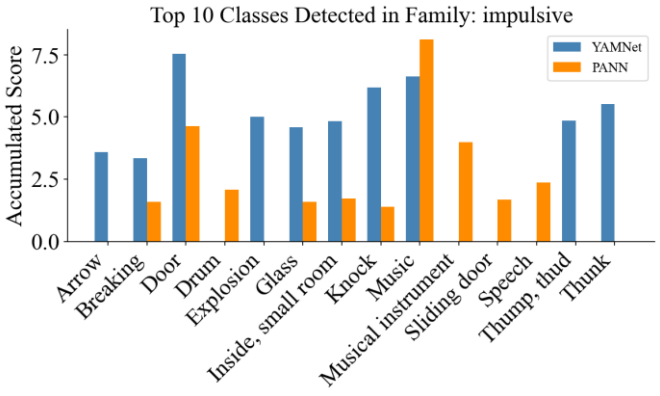
	YAMNet	PANN (CNN14)
Maximum CPU usage of Raspberry Pi 4	12%	90%*
Classification time on Raspberry Pi 4*	~200 ms	150 ms
Maximum memory on Raspberry Pi	850 MB	**

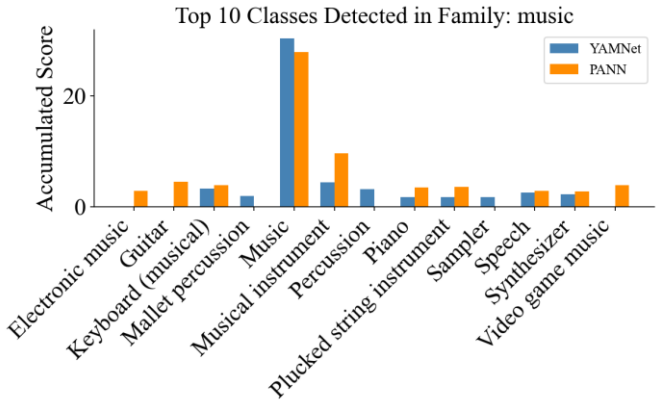
*Tested on one CPU core only
** not tested

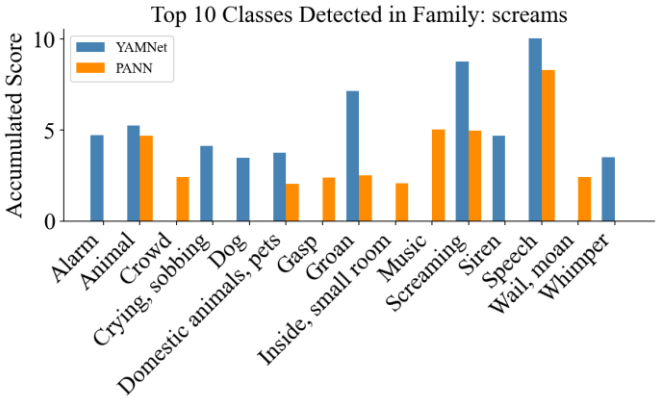
IV. Database Construction (HospitalDB)

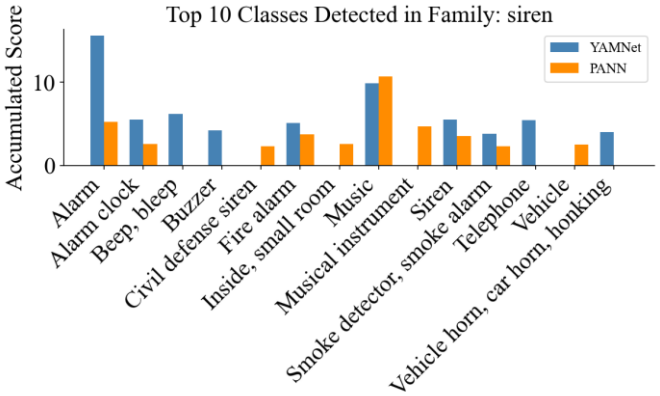
In order to validate the models, a second dataset, the HospitaldB dataset, was created from the scratch only with real sounds from a hospital environment. This dataset consists of approximately 50 sounds from each of the following 10 categories (families): speech (hallway speech, consultations with patients, doctor and nurse conversations), screams (screams, cries, moans), alarm (monitor alarms, emergency devices), telephone (landline and mobile phone rings), music (background music in waiting rooms or bedrooms), whistle (whistles, unidentified short beeps), wheels (sounds of stretchers, wheelchairs, and support carts), snore (snoring sounds, deep breathing), waterfall (running faucets, showers, liquid sounds), and impulsive (falling objects, thuds, bangs).

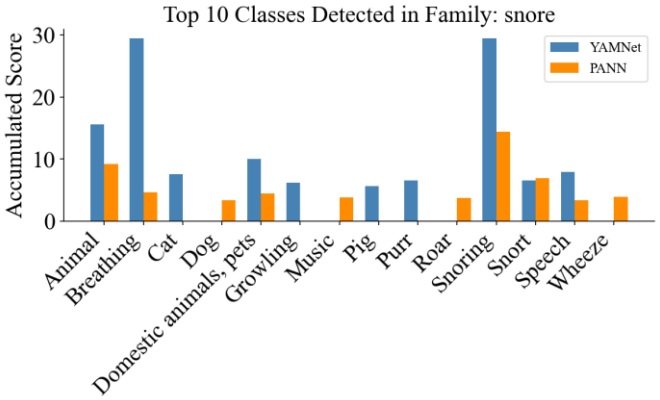
To check whether the models worked with the HospitaldB dataset, that is, whether the dominant classes returned from the models YAMNet ans PANN (523 classes) corresponded to the categories (families) defined in HospitaldB, the Accumulated Score results for the various families were made, as depicted in Figure 4.

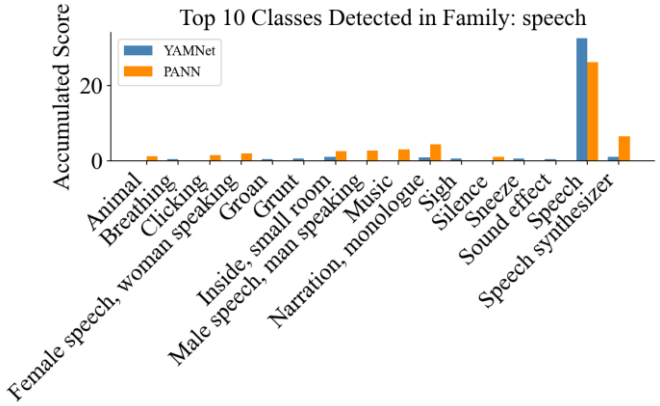












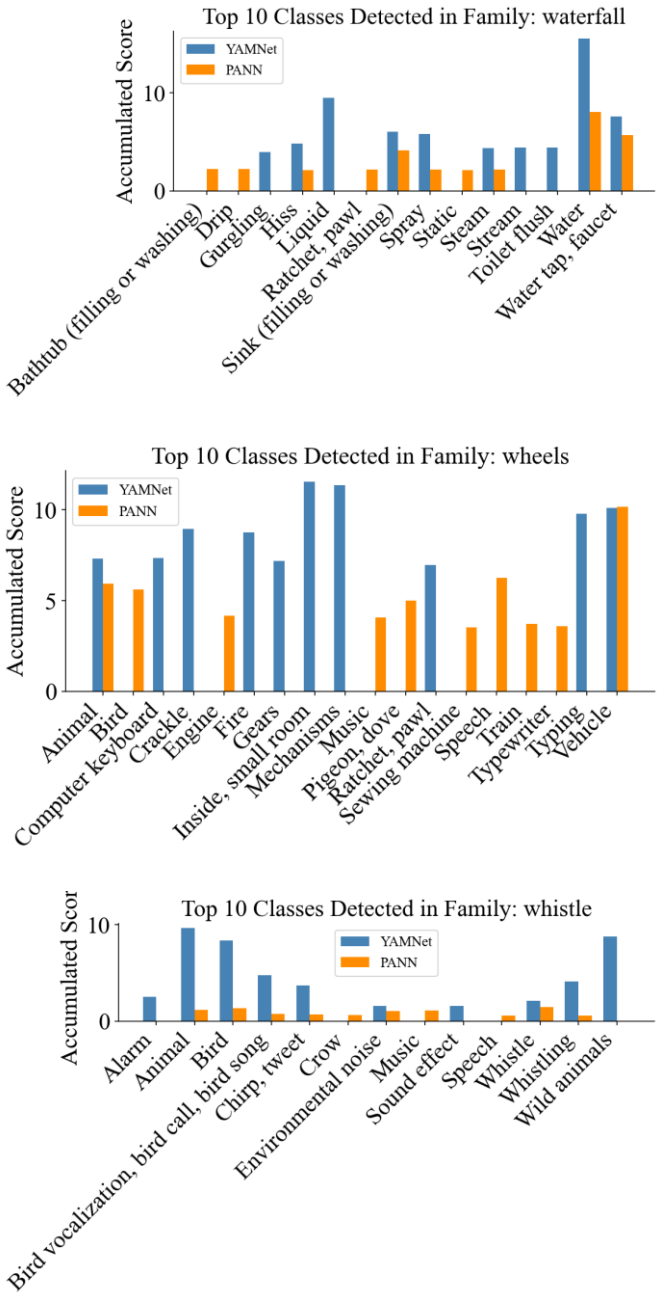


Fig. 4. Accumulated Score results for the various families defined in HospitaldB dataset.

The results were almost consistent using the models studied. In fact, most of the classes returned by the models correspond to the type of sound events of each family described in the HospitaldB dataset. Thus, the models were validated, being good candidates to the project.

V. Implementation

The system performs efficiently on devices like the Raspberry Pi 4 B+, 1 GB, and stores results with prediction confidence scores. Additionally, a mapping was created between the 527 AudioSet classes and 10 semantic families relevant to hospital environments, enabling easier human interpretation of the acoustic context.

The YAMNet model showed the best classification performance for this project (and was used in the final system). This model performs inference on 1-second audio segments at 16 kHz, converting samples into mel-scaled spectrograms, and produces a distribution of scores for 527 classes. The scores are values from 0 to 1 distributed, with the number of different values equal to the number of classes. The score at each index of the list corresponds to the score of the class of the same index in AudioSet, YAMNet Google Research (2020) [7].

The main advantages of YAMNet:

- Lightweight and efficient (suitable for local execution);
- Well-trained with real examples from YouTube;
- Easily integrateswith the Python Tensorflow library (version 2. x).

YAMNet is used in a customized way: instead of considering all 527 class scores, we implemented family regrouping logic and filtering irrelevant classes (e.g., "Silence"). Additionally, a second dataset, as mentioned before, the HospitaldB dataset, was created only with real sounds from a hospital environment, in order to validate the YAMNet model.

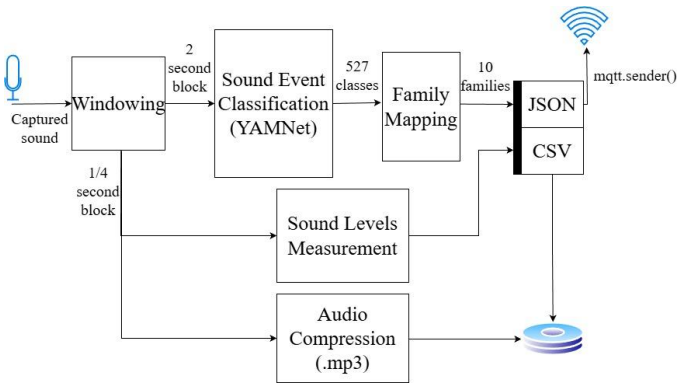
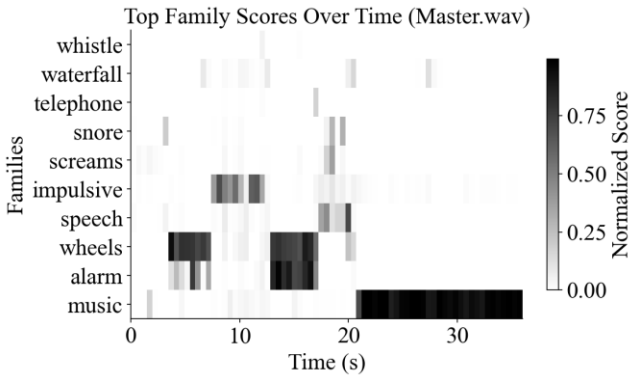


Fig. 5. Magnetization as a function of applied field.

The results are depicted in Figure 6 for Master.wav sounds and Figure 7 for hospitaldb_merged.wav (compilation of all HospitaldB dataset sounds).



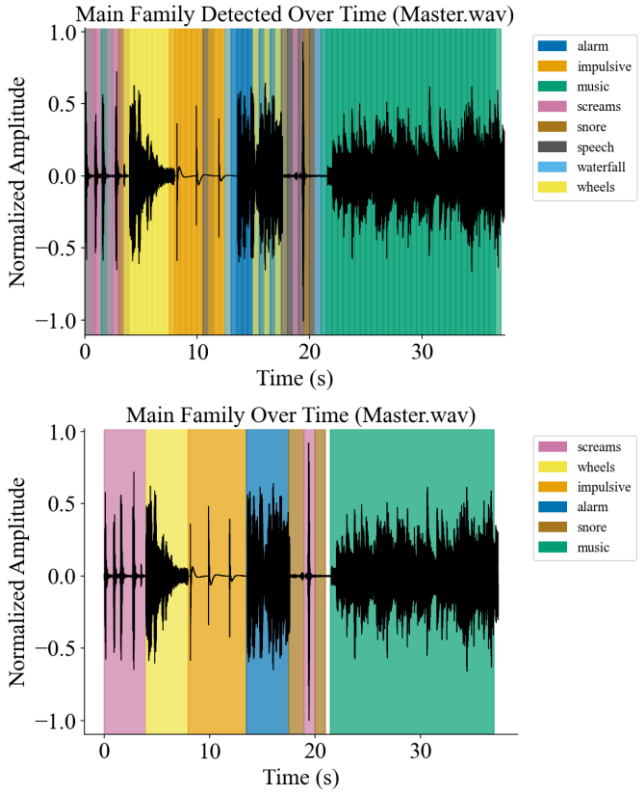
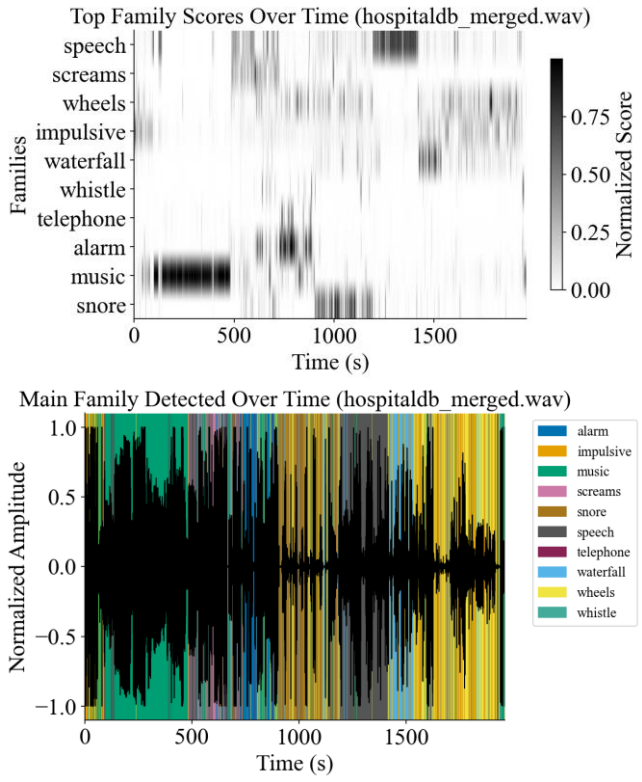


Figure 6. Classification test performed on the set of sounds from the HospitaldB dataset. The top image shows the audio signal waveform, marking the zones corresponding to each family with different colors. The bottom image shows the classification results, showing the classification scores.



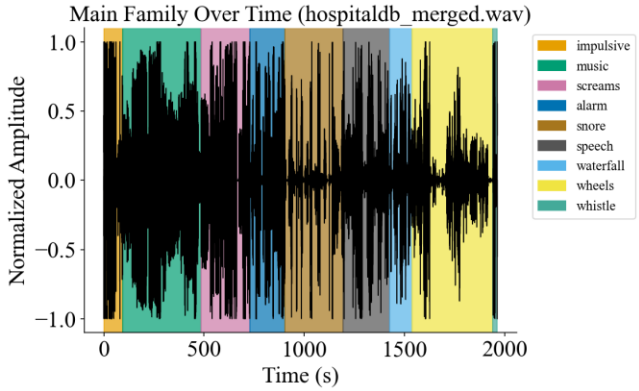


Figure 7. Classification test performed on the set of sounds from the HospitaldB dataset. The top image shows the audio signal waveform, marking the zones corresponding to each family with different colors. The bottom image shows the classification results, showing the classification scores.

The classification test performed on the set of sounds from the HospitaldB dataset (approximately half an hour of audio). The YAMNet model performs reasonably well, successfully identifying virtually all sound events. However, additional testing is needed to improve the accuracy of the classification result using the Transfer Learning approach.

VI. Conclusion and Future Work

The main objective of this project was the detection and classification/identification of relevant sound events usually occurring in a hospital environment, with a focus on capacity real-time execution and integration with existing monitoring systems.

Based on this objective, a complete pipeline was developed that allows capturing, classifying and interpreting sound events through the model YAMNet, with an additional semantic layer of aggregation by families and a filtering logic adapted to the hospital context.

The main message to take away from this work is the observation that models generalist audio classification systems, such as YAMNet, can be successfully adapted to specific domains, just by changing the numbers and family names. The HospitaldB database, class mapping for families and the introduction of relevance weights are central elements to achieve this objective.

Furthermore, the project revealed the importance of iterative testing and validates operations at multiple levels: from testing with long sound files and complexes, up to short-term block inference analysis. This approach allowed us to verify that, although models such as PANNs have robust performance in noisy environment, it requires more computational resources (memory and CPU) which further limits its application in IoT devices.

Another relevant contribution of the project was experimentation with different score aggregation strategies, including the exclusion of irrelevant classes (e.g., ”Silence”) and weighting based on statistical analysis of real data.

These strategies have proven to be fundamental for improving accuracy and usefulness of classification results.

The successful execution of the classifier on Raspberry Pi,

with integrated multiprocessing and sending via MQTT [17], confirms the viability of the solution for implementation in real scenarios. The modularity of the code, with clear separation between capture, classification and communication, facilitates adaptation to other environments, such as schools, factories or public spaces.

This project, as built, offers multiple opportunities for continuity and reuse:

The HospitaldB database can be expanded and used as a reference dataset for new models.

The mapping and weighting system can be adapted to other semantic domains.

The classification pipeline can serve as a basis for monitoring systems acoustic control in real time.

As a future developments we need to enhance the YAMNet model, using Transfer Learning to improve the classification of classes that currently have less training.

The inclusion of context detectors for dynamic adjustment of weights and families according to the hospital area (e.g., emergency, inpatient, UCI, will be another improvement to be made.

ACKNOWLEDGMENT

The authors would like to express their gratitude to André Rocha (ESELx, IPL, Fablab Benfica) for support in the 3D fabrication and printing of project components.

We also acknowledge the Laboratory of Electronics of CEDET – Centro de Estudos e Desenvolvimento de Electrónica e Telecomunicações (ISEL, IPL), for providing facilities, equipment, and technical assistance throughout the electronic design and testing phases of this work.

Special thanks are extended to trainee João Brilhante, (DUAL professional school), for his valuable contribution in the design of mechanical parts and printed circuit boards.

REFERENCES

[1] Preto Paulo, J. et al., "SOUNDMETERHOSP - A Platform For Aggregating Sound Events In Hospitals," in 3º Simpósio de Acústica e Vibrações, 2025, 7 november, Coimbra, Portugal.

[2] Preto Paulo J., Coelho J., Figueiredo, M., Statistical classification of road pavements using near field vehicle rolling noise measurements. J Acoust Soc Am. 2010 Oct;128(4):1747-54. doi: 10.1121/1.3466870. PMID: 20968348.

[3] Mendes, A., J., Trigo, P., Preto Paulo, J., Hyperparameter Optimization for the Entire Classification Process of Impulsive Sounds. Proceedings of the ICAART - International Conference on Agents and Artificial Intelligence. 24-26 de February de 2024, Rome, Italy;

[4] Alqudaihi KS, Aslam N, Khan IU, Almuhaideb AM, Alsunaidi SJ, Ibrahim NMAR, Alhaidari FA, Shaikh FS, Alsenbel YM, Alalharith DM, Alharthi HM, Alghamdi WM, Alshahrani MS. Cough Sound Detection and Diagnosis Using Artificial Intelligence Techniques: Challenges and Opportunities. IEEE Access. 2021 Jul 15;9:102327-102344. doi: 10.1109/ACCESS.2021.3097559. PMID: 34786317; PMCID: PMC8545201.

[5] SONYC (Sounds of New York City) https://wp.nyu.edu/sonyc/

[6] Paulo, J., et. al., " Framework to Monitor Sound Events in the City Supported by the Fiware Platform ". TecniAcustica2015, Valencia, Spain, 21-23 Novembro (2015). Proceedings of TecniAcustica2015. ISBN: 978-84-87985-26-3 ISSN: 2340-7441 (Versión Digital)

[7] Preto Paulo, J., Alves, J., Marques, G., Guerreiro, P., A Low-Cost Sound Event Detection and Identification System for Urban Environments, i-ETC: ISEL Academic Journal of Electronics Telecommunications and Computers, Vol. 6, n. 1 2020. ISSN: 2182-4010. http://dx.doi.org/10.34629/ipl.isel.i-ETC.61

[8] G. R. (2020), "YAMNet: Audio event classifier based on MobileNet." [Online]. Available: https://github.com/tensorflow/models/tree/master/research/audioset/.

[9] Google Research. (2020). YAMNet: Audio event classifier based on MobileNet, https://github.com/tensorflow/models/tree/master/research/audioset/

[10] Gemmeke, J. et. al., AudioSet: An ontology and human-labelled dataset for audio events, ICASSP 2017, https://github.com/DTaoo/VGGish

[11] Aurora Cramer, Ho-Hsiang Wu, Justin Salamon, and Juan Pablo Bello. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), pages 3852–3856, Brighton, UK, May 2019 - OpenL3: https://github.com/marl/openl3

[12] Hershey, S. et. al., CNN Architectures for Large-Scale Audio Classification, ICASSP 2017.
Gemmeke, J. F., Ellis, D. P. W., Freedman, D., Jansen, A., Lawrence,W., Moore, R. C., Plakal, M., Ritter, M. (2017). AudioSet: An ontology and human-labeled dataset for audio events. IEEE ICASSP.

[13] Kong, Q., Cao, Y., Iqbal, T., Wang, Y., Wang, W., Plumbley, M. D. (2020). PANNs: Pretrained Audio Neural Networks. Available at: https://github.com/qiuqiangkong/audioset tagging cnn

[14] Urbansound8k ecapa - Sound Recognition https://dataloop.ai/library/model/speechbrain_urbansound8k_ecapa/

[15] Salamon, J., Jacoby, C., Bello, J. P. (2014). UrbanSound8K Dataset. Available at: https://urbansounddataset.weebly.com/urbansound8k.html

[16] OASIS Standard. (2014). MQTT Version 3.1.1. Available at: http://docs.oasisopen. org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html

[17] TensorFlow Authors. (2023). TensorFlow: An end-to-end open source machine learning platform. Available at: https://www.tensorflow.org

[18] McFee, B., Raffel, C., Liang, D., Ellis, D. P. W., et al. (2022). Librosa: Python package for music and audio analysis. Available at: https://librosa.org