RESEARCH & INNOVATION FOUNDATION

2016-2020 RESTART SEARCH

ΠΡΟΓΡΑΜΜΑΤΑ ΕΡΕΥΝΑΣ, ΤΕΧΝΟΛΟΓΙΚΗΣ ΑΝΑΠΤΥΞΗΣ ΚΑΙ ΚΑΙΝΟΤΟΜΙΑΣ

# THE RESEARCH & INNOVATION FOUNDATION PROGRAMMES FOR RESEARCH, TECHNOLOGICAL DEVELOPMENT, AND INNOVATION RESTART 2016 – 2020

Green-HIT

| | |
|---|---|
| **Pillar** | I. Smart Growth |
| **Programme** | CO-DEVELOP |
| **Project Acronym** | Green-HIT |
| **RIF Project Number** | CODEVELOP-ICT-HEALTH/0322/0135 |
| **Proposal Title** | A Green - Holistic IoT platform for Forest Management and Monitoring |
| **Project Coordinator** | Frederick Research Center (FRC) |
| **Work Package Number** | WP6 |
| **Work Package Title** | Green-HIT Intelligence Modules |
| **Deliverable Number** | D6.2 |
| **Deliverable Title** | Detection of Illegal Logging and Hunting Module (Prototype & Report) |

| **Dissemination level** | | |
|---|---|---|
| PU | Public | X |
| CO | Confidential, only for members of the consortium (including RIF) | |

Funded by the European Union NextGenerationEU

Cyprus tomorrow RECOVERY AND RESILIENCE PLAN

Republic of Cyprus

## AUTHORS


| Author | Institution | Contact (e-mail, phone) |
|---|---|---|
| Andreas Pamboris | **FRC** | res.ap@frederick.ac.cy |
| Andreas Constantinides | **FRC** | com.ca@frederick.ac.cy |


## DOCUMENT CONTROL

| Document version | Date | Change |
|---|---|---|
| v0.1 | 30/04/2025 | Draft |
| v1.0 | 26/05/2025 | Final Version |

## Executive Summary

This deliverable outlines the development and implementation of a sound-based detection system designed to identify illegal logging and hunting activities through acoustic monitoring. Recognizing that such illicit actions often produce distinctive audio cues (such as chainsaws and gunshots), the system employs an AI-driven approach to automatically detect and classify relevant environmental sounds in real-time. This module was developed as part of WP6.

# Table of Contents

# 1. Introduction

Illegal logging and hunting present significant threats to biodiversity, forest ecosystems, and local economies. These unlawful activities not only cause irreversible damage to the environment but also undermine conservation efforts and sustainable development initiatives. In many cases, these events occur in remote or protected areas where manual monitoring is infeasible or insufficient. To address these challenges, the Green-HIT project introduces an integrated sound-based detection solution capable of identifying illicit activities using both in-situ edge-based and cloud-hosted analytical pipelines.

Deliverable D6.2 details the development of an AI-powered acoustic monitoring system that can detect sounds characteristic of illegal logging and hunting (such as chainsaw noise or gunshots) with high accuracy and real-time responsiveness. The module is implemented in two complementary configurations: (1) low-power embedded modules for in situ deployment in forested regions, and (2) a cloud-based architecture that supports scalable processing and richer post-analysis. Both configurations use advanced signal processing techniques and leverage deep learning architectures, including 1D CNNs and YAMNet, to classify environmental audio events effectively. This deliverable outlines the architecture, implementation, evaluation, and rationale behind the design decisions, demonstrating how Green-HIT's detection module enhances forest monitoring capabilities across diverse operational scenarios.

The remainder of this deliverable is structured as follows:

- Section 2 outlines the core objectives of the illegal logging and hunting detection module and defines the expected operational outcomes.
- Section 3 presents the in-situ edge-based detection approach, including hardware setup, audio preprocessing pipeline, model architecture, training methodology, and performance metrics.
- Section 4 details the cloud-based approach, including the use of YAMNet, preprocessing pipeline, inference process, classifier training, and evaluation using the ESC-10 dataset.
- Section 5 explains the rationale behind key design and implementation choices across both deployment modes.
- Section 6 concludes the deliverable by summarizing outcomes, practical implications, and directions for future enhancements.

## 2. Objectives of Illegal Logging and Hunting Detection Module

The primary aim of the *Illegal Logging and Hunting Detection* module is to develop a sound-based detection system that can effectively identify and classify audio events related to illegal logging and hunting. The specific objectives include:

- To design and implement a robust pipeline for real-time or near-real-time processing of environmental audio data.
- To utilize pre-trained machine learning models to minimize the need for extensive training on domain-specific datasets.
- To enable modular deployment on edge or cloud systems, adaptable to various geographical locations and acoustic environments.
- To demonstrate high classification accuracy using standard environmental sound benchmarks such as ESC-10.

# 3. In-Situ Edge-Based Approach

The in-situ detection module (Figure 4) for detecting logging and hunting sounds (e.g., chainsaws and gunshots) is deployed in areas such as *Orkonta*. It is designed to function in harsh environmental conditions and relies on DSP-equipped microcontrollers. This module's hardware is detailed in deliverables D4.1 and D4.2.

**Figure 4:** In-situ audio recognition module installed deeper inside the forested areas.

The DSP module of the board comes with pre-installed audio recognition models. Initial pre-installed models were, however, replaced with custom-trained models using the *Edge Impulse* platform due to insufficient accuracy. The training datasets included *gunshot* and *chainsaw* sounds along with *environmental noises* like *birds* and *wind*, obtained from sources such as ESC-50 and Kaggle. The total duration of all data samples used is 1h 15m 13sm, with different samples having variable lengths. Audio pre-processing was performed using Mel-filterbank Energy (MFE), which is well-suited for non-speech data.

Models were trained using a 1D CNN architecture with dropout to mitigate overfitting. The illegal logging and hunting model achieved 98.1% accuracy for chainsaws and 91.2% for gunshots. Deployed on Nordic nRF52840 modules with low resource usage (RAM: 20.8kB, Flash: 51.8kB), the system continuously listens and sends alerts via LoRa when classification scores exceed 99.5%. Once an alert is received, a UAV is dispatched to validate the situation, minimizing false alarms and ensuring timely response.

Field tests showed that gunshots could be detected across wide ranges, while chainsaw sounds were more vulnerable to terrain-related obstructions. To counter this, multiple audio modules may be required in rugged terrains or mounted on elevated positions for optimal coverage.

## 3.1 Signal Processing Approach

The MFE extracts a spectrogram from the audio signals provided in the dataset using time and frequency features in a non-linear scale called the Mel-scale. The MFE was set with the following parameters:

- Frame Length: 0.02 seconds
- Frame Stride: 0.01 seconds

- Filter Number: 40
- Fast Fourier Transform (FFT) Points: 512
- Low Frequency Band: 300Hz
- Noise Floor: -75dB

First a spectrogram was created using the Frame Length, Frame Stride and FFT points provided. It divides the signal window of the data into multiple overlapping frames based on the Frame Length and Stride provided. For example, a sample with a window of 1 second (using the above parameters) would create 99 timeframes. An FFT is then calculated for each frame. The number of frequency features is equal to the FFT points divided by two (2) plus one (1). The Noise Floor is then applied to the spectrum.

After the spectrogram is computed, the triangular filters are applied on a Mel-scale to extract frequency bands, using the Low Frequency Band parameter as low and zero as high. The number of frequency features extracted is determined by the Filter Number parameter.

The FFT Bin Weighting graph (Figure 5) shows how the FFT bins are scaled and summed into the output columns based on the parameters above.

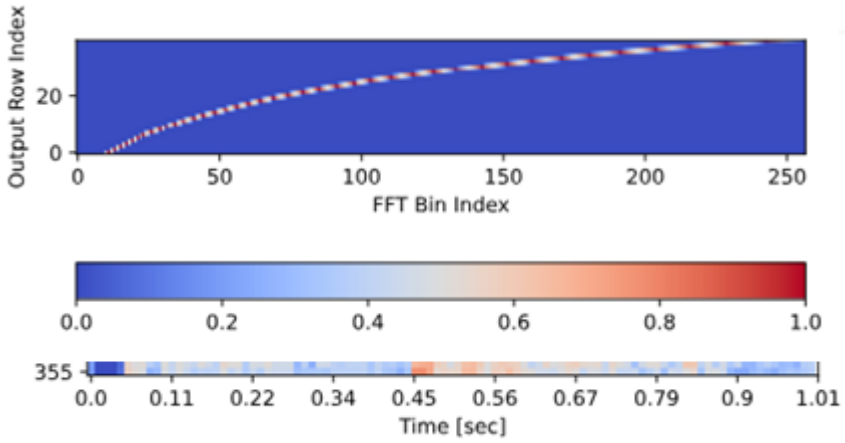



**Figure 5:** FFT Bin weighting graph.

Figure 6 demonstrates the DSP output of the MFE on a gunshot audio sample.

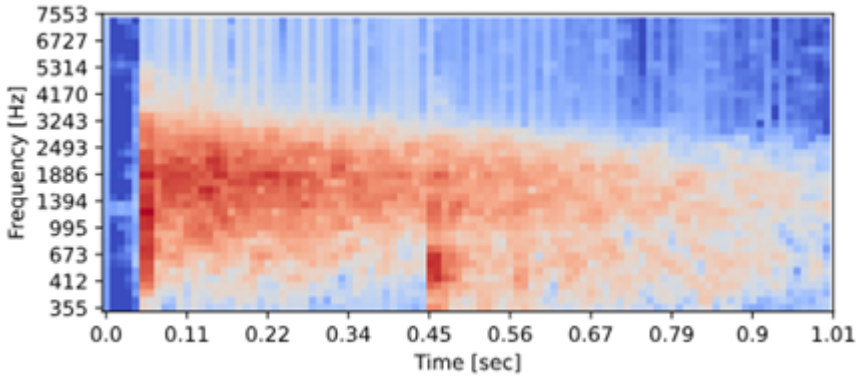



**Figure 6:** DSP output.

## 3.2 Model Training and Neural Network Architecture

Model training was done using the Classification (Keras) learning block while utilizing a One-Direction Neural Network architecture. This was chosen because it is suitable for two-dimensional data like audio. The following settings were used to train the model:

- Training Cycles: 200
- Learning Rate: 0.0005

The training algorithm passed through the training data 200 times (200 training cycles) and adapted the model's parameters at the set learning rate. Training cycles and learning rates were determined by considering several models with different training settings to find those exhibiting the best accuracy while avoiding overfitting. The network used the following neural network architecture.

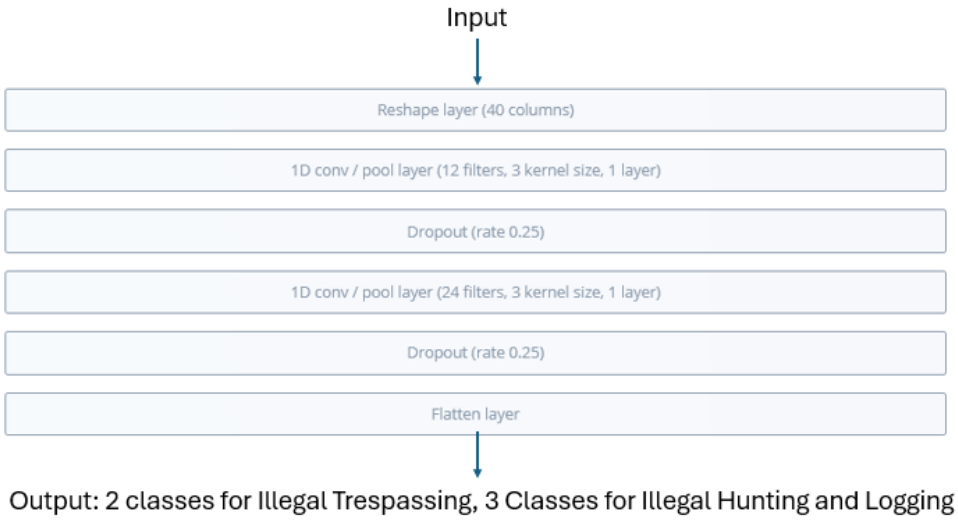



**Figure 7:** Neural Network architecture.

1. The inputs are the extracted features taken during signal processing; these features pass through each layer of the above architecture.
2. The reshape layer turns the one-dimensional data from the feature into multi-dimensional data to feed into the convolutional layer.
3. The data is then passed through two (2) convolutional layers:
   - the first slides 12 filters across the sequence with three (3) kernel size that moves at one (1) step at a time
   - the second slides 24 filters.
4. After each convolutional layer, several network connections are cut from the model to reduce overfitting with a dropout probability set at 25%.
5. The features are then flattened back into a single dimension to provide the output, which is separated in three (3) classes for illegal hunting and logging (Chainsaw, Gun, and Other).

## 3.3 Model Accuracy and Performance

This section discusses the detection accuracy of the developed models.

For the illegal hunting and logging detection module (Figure 8), *Chainsaw* was correctly classified at 98.1% while the other 1.9% were classified as *Other*. *Gunshot* was correctly classified at 91.2%, while 7% of Gunshot samples were incorrectly classified as *Chainsaw,* and another 1.8% were classified as *Other*.

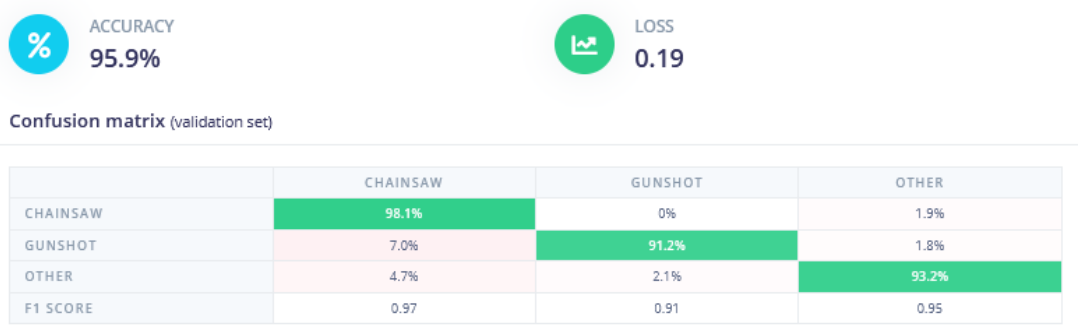



**Figure 8:** Illegal hunting and logging detection model - confusion matrix.

The performance metrics of the model are the following:

- Inferencing Time: 7ms
- Peak Ram Usage: 20.8k
- Flash Usage: 51.8k

Considering the Nordic nRF52840 module inside the Audio Recognition Module has 1MB flash and 256KB RAM, the model is well within the acceptable performance requirements to function.

## 3.4 Model Implementation

The developed models were deployed as an Arduino library to perform inference continuously in the field. The models were tested with the audio samples used in the dataset and real-life scenarios, such as real vehicle engines, chainsaws, and gunshots. Due to hardware bandwidth constraints, the sound quality of real-life experiments was lower, resulting in a lower detection accuracy under real conditions.

While inferencing, when the classification exceeds a certain threshold, e.g., 99.5% for chainsaw (Figure 9), the device sends a LoRa payload (Figures 10 and 11) to the platform that indicates what sound was detected.

```
else if(result.classification[ix].label == "Chainsaw" && result.classification[ix].value>0.995)
{
    Serial.println("Chainsaw");

    led_state = !led_state;
    digitalWrite(LED_BLUE, led_state);
    digitalWrite(LED_GREEN, led_state);
    int id_case = cmd_id;
    //    play_task();
    xQueueSend(integerQueue, &id_case, portMAX_DELAY);
    int_flag = 0; //clear the interrupt flag
    if(chain_flag==0){
        chain_flag=1;
    }
```

**Figure 9:** Chainsaw classification threshold.

```
Predictions (DSP: 32 ms., Classification: 47 ms., Anomaly: 0 ms.):
    Chainsaw: 0.99609
Chainsaw
lmh_send ok count 15
    Gunshot: 0.00000
    Other: 0.00391
```

**Figure 10:** Chainsaw sound detection and LoRaWAN transmission.

| | UL/DL | FCNT | Timestamp | Content | LoRaWAN™ Port | RSSI | SNR | ESP | SF |
|---|---|---|---|---|---|---|---|---|---|
| ^ | ↗ | 31 | Today 17:09:22 | DATA | 2 | -68 dBm | 14.25 dB | -68.16 dBm | SF7 |

| | |
|---|---|
| Model Identifier: | genericlora:1 |
| Protocol Identifier: | Not Available |
| Encoded Payload: | 02 (not encrypted) |

**Figure 11:** Chainsaw payload sent to network server.

# 4. Cloud-Based Approach

## 4.1 Model Selection and Architecture

YAMNet (Yet Another Mobile Network)[1] was selected as the core component of the audio detection module due to its proven effectiveness in environmental sound classification and its broad training on Google's AudioSet[2]. AudioSet is a large-scale dataset comprising over two (2) million 10-second audio clips extracted from YouTube videos and annotated with labels from an ontology of 527 sound event classes. This dataset spans a wide variety of human, musical, animal, and environmental sounds, giving YAMNet the ability to generalize across diverse real-world acoustic scenarios.

At the heart of YAMNet is MobileNetV1[3], a lightweight convolutional neural network architecture optimized for resource-constrained environments such as mobile and embedded devices. Unlike traditional convolutional networks, MobileNetV1 utilizes depth-wise separable convolutions, which decompose standard convolutions into two simpler operations: depth-wise convolutions (which apply a single filter per input channel) and point-wise convolutions (1x1 convolutions to combine the outputs). This significantly reduces the number of parameters and computational cost without substantially sacrificing accuracy, making the architecture ideal for edge applications in remote areas where power and processing capabilities are limited[4,5].

By leveraging YAMNet, the module benefits from both high classification accuracy and real-time responsiveness, essential for the detection of transient and sporadic audio events such as chainsaw sounds, gunshots, and vehicle noises commonly associated with illegal logging and hunting activities.

## 4.2 Audio Preprocessing Pipeline

Before audio signals can be analysed by YAMNet, they must be pre-processed to ensure compatibility and maximize model performance. The preprocessing pipeline includes several key steps, described next:

- Resampling: All incoming audio is resampled to a standard rate of 16 kHz and converted to mono. This

[1] YAMNet model page**:** https://tfhub.dev/google/yamnet/1

[2] AudioSet Dataset: https://research.google.com/audioset/

[3] Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., & Ritter, M. (2017). Audio Set: An ontology and human-labeled dataset for audio events. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 776-780).

[4] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[5] Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., ... & Saurous, R. A. (2017). CNN architectures for large-scale audio classification. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 131-135).

ensures consistent input formatting and reduces variability in audio quality.

- Framing: Audio streams are divided into overlapping frames of 0.96 seconds with a hop length of 0.48 seconds. This step helps maintain temporal resolution and provides context for transient audio events.
- Spectrogram Computation: Each frame undergoes a *Short-Time Fourier Transform (STFT)* to convert the time-domain signal into frequency-domain data, capturing changes in pitch and tone over time.
- Mel Spectrogram and Log Scaling: The STFT output is converted into a Mel spectrogram using a 64-band filterbank, followed by logarithmic scaling. This process mimics the human ear's perception of sound and compresses the dynamic range for improved analysis.

## 4.3 Model Inference

After the audio signal undergoes the full preprocessing pipeline, i.e., resampling, framing, Fourier transformation, Mel scaling, and logarithmic compression, the resulting log Mel spectrograms serve as the direct input to the YAMNet model. These spectrograms represent the time-frequency characteristics of the audio signal in a form that is both biologically meaningful (mimicking human auditory perception) and well-suited for machine learning models.

Upon receiving these inputs, YAMNet performs frame-wise inference, typically at a resolution of approximately 0.96 seconds per frame. For each frame, the model produces two distinct outputs:

**1. Class Scores (Logits):** These are floating-point values representing the model's confidence levels across a predefined set of 521 audio event classes, derived from the AudioSet ontology. Each value corresponds to a specific class, such as "chainsaw," "gunshot," "car engine," or "animal call." A higher score indicates greater confidence that the class is present in the frame.

These scores can be used for direct sound event classification, where the system reports the top-N most likely sound classes. For practical deployment, thresholding or SoftMax normalization is applied to convert logits into probabilities.

**2. Embedding Vector (1024-dimensional):** This vector represents a high-level, compact semantic encoding of the frame's acoustic content. Unlike class scores, which tie directly to known labels, the embedding captures nuanced spectral-temporal features that characterize the sound holistically.

These embeddings are highly versatile and can be used for a wide range of downstream tasks, including:

- Classification using simpler supervised models (e.g., logistic regression or SVM), particularly effective when ground truth labels are known for a new task.
- Clustering to identify recurring patterns or group similar sound events (e.g., all variations of chainsaw noises).
- Anomaly Detection, where embeddings from expected background sounds (e.g., wind, bird calls) are used to train a normal profile, and deviations suggest suspicious events.

Since real-world audio recordings typically span several seconds to minutes, and YAMNet operates on roughly 1-second frames, an aggregation strategy is necessary to derive a fixed-size representation of the entire audio clip. The most common technique is mean pooling, where all frame embeddings are averaged to produce a single 1024-dimensional vector summarizing the entire clip. This method:

- Maintains simplicity and computational efficiency.
- Works well when the relevant sound event is prominent or spans multiple frames.

More advanced strategies could include:

- Weighted averaging based on class scores.
- Attention-based pooling, which learns to focus on the most informative segments.
- Temporal modeling, using RNNs or Transformers to retain time dynamics across frames.

This dual-output nature of YAMNet, offering both interpretable class scores and rich, abstract embeddings, makes it a powerful tool for environmental audio analysis, particularly in applications like illegal logging and hunting detection where both known event recognition and unusual pattern detection are critical.

### 4.4 Training and Evaluation Methodology

To assess the module's performance, an extensive evaluation was conducted using the ESC-10 dataset (subset of the larger ESC-50 dataset[6]), a benchmark collection of environmental sounds across 10 categories. This dataset serves as a reliable proxy for assessing real-world audio classification tasks.

The evaluation process followed these steps:

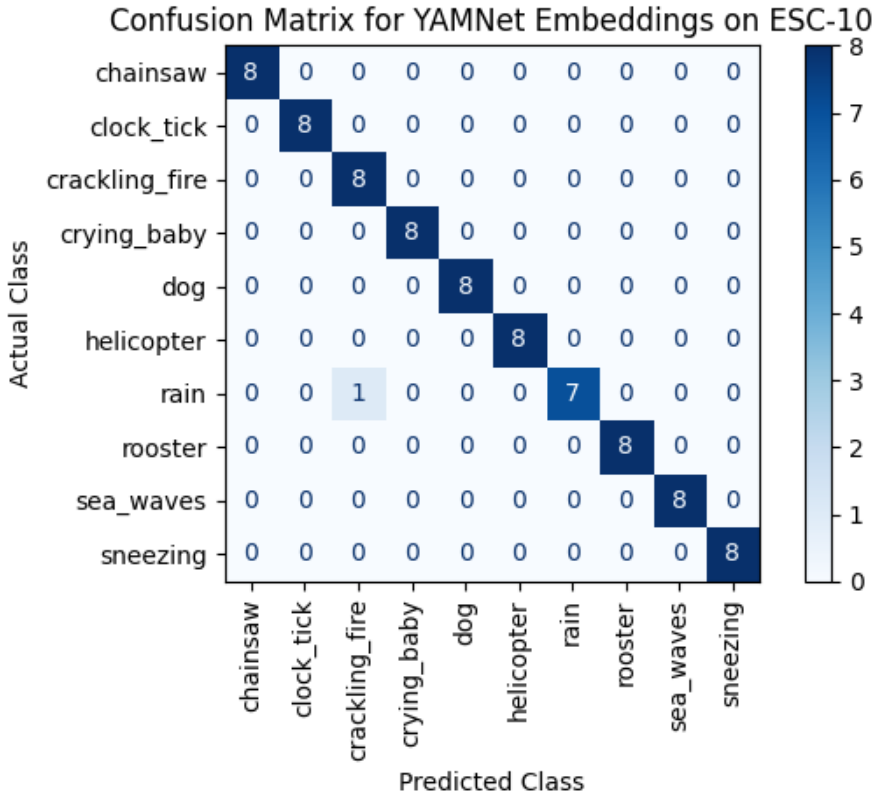
1. Dataset Preparation: The ESC-10 dataset was loaded. Each audio clip was pre-processed to meet YAMNet's input requirements, including resampling to 16 kHz mono audio.
2. Feature Extraction: The pre-trained YAMNet model was used to extract 1024-dimensional embedding vectors for each audio clip in the ESC-10 dataset. For each clip, the embeddings from all frames were aggregated by taking their mean to obtain a single, fixed-size representation per audio file.
3. Data Splitting: The dataset of extracted YAMNet embeddings and corresponding ESC-10 class labels was split into *training* and *test* sets. A standard 80/20 split was used, with stratification to ensure that the class distribution was maintained in both the training and testing subsets.
4. Classifier Training: A simple Logistic Regression classifier was trained on the YAMNet embeddings from the training set. This step demonstrates the effectiveness of YAMNet's embeddings as features for a downstream classification task on a new dataset.
5. The evaluation focused on two (2) key metrics:
    - **Overall Accuracy:** The proportion of correctly classified audio clips in the test set.
    - **Confusion Matrix:** A 10x10 matrix visualizing the classification performance for each of the 10 ESC-10 classes. The confusion matrix shows the number of true positives, false positives, false

[6] https://github.com/karoldvl/ESC-50

negatives, and true negatives for each class, providing a detailed breakdown of where the model succeeds and where it makes errors.

## 4.5 Evaluation Results

The classifier achieved a test accuracy of **98.75%**, underscoring the effectiveness of YAMNet embeddings. The corresponding confusion matrix (provided below) has revealed minimal misclassifications, which validates the model's reliability for this domain.




## 4.6 Integration Strategy

The audio detection module was designed for seamless integration into broader environmental monitoring frameworks. Depending on deployment needs and available infrastructure, it can operate either on edge devices (e.g., Raspberry Pi, microcontrollers with AI accelerators) or be cloud hosted. Audio sensors in forested or protected areas can record data continuously or on a scheduled basis, which is then streamed or uploaded for analysis.

The modular design ensures that updates to the model or preprocessing pipeline can be implemented without disrupting existing installations. Moreover, the system can be expanded to include other modalities, such as image or video detection, to provide a multi-sensory fusion-based monitoring capability.

## 5. Rationale Behind Choices Made

Several deliberate decisions were made during the development of both the in-situ and cloud-based detection modules to balance accuracy, efficiency, and deployability in real-world scenarios. Both approaches rely on acoustic sensors, which provide round-the-clock monitoring and do not require line-of-sight, making them suitable for wider area surveillance and detection of distant or obscured events. These sensors are less affected by terrain and lighting but may suffer from environmental noise or audio muffling depending on placement. This makes elevation and redundancy essential.

For the in-situ edge-based approach, the use of the *Edge Impulse* platform for embedded AI model development was based on its rapid prototyping capabilities, seamless deployment on microcontrollers, and suitability for low-power environments. It enabled the design of efficient neural networks with short inference times and minimal memory usage, making them ideal for Nordic nRF52840 microcontrollers used in the field.

For the cloud-based approach, YAMNet was chosen due to its extensive training, robust architecture, and proven track record in diverse audio classification tasks. Additionally, the model's compatibility with low-resource hardware makes it suitable even for field deployment where high-performance servers may not be available. Log Mel spectrograms were selected as the primary input format due to their biological plausibility and effectiveness in compressing audio features. This transformation ensures that subtle but significant acoustic cues, such as distant gunshots or chainsaw noise, are preserved. The use of logistic regression for downstream classification was a strategic choice to demonstrate that the extracted embeddings are linearly separable and informative. This approach minimizes training overhead and allows quick adaptation to new environments or label sets with minimal retraining. Finally, ESC-10 was selected as a benchmark dataset because of its relevance to environmental sound classification and the availability of well-documented evaluation metrics.

## 6. Conclusions

Deliverable D6.2 presents an end-to-end solution for the detection of illegal logging and hunting activities using environmental sound analysis, implemented through both in situ and cloud-based infrastructures. The in-situ module, based on Edge Impulse-trained 1D CNNs, demonstrates lightweight yet robust classification of chainsaws and gunshots under real-world constraints. The cloud-based module, leveraging YAMNet and ESC-10 embeddings, achieves a test accuracy of 98.75%, showcasing its effectiveness in high-performance classification tasks.

Together, these modules form a complementary system that provides flexible, scalable, and accurate detection of acoustic signatures associated with forest crimes. Their deployment enables proactive environmental surveillance, supports timely response efforts, and contributes to long-term biodiversity protection strategies. Future work will focus on extending the dataset with locally sourced sounds, refining detection in challenging terrains, and integrating multi-modal sensing capabilities such as image-based and drone-assisted verification systems.