Final Report
Détection de feux de forêt - Wildfire detection

Pôle Intelligence Artificielle
P10 S7 05 - CentraleSupélec

Karina Musina
Noé Bertramo
Erwin Deng
Jad Tahri
Elias Al Bouzidi

Client:

Frédéric Magoules

Supervisors:

Wassila Ouerdane
Jean-Philippe Poli

February 6, 2024

Contents

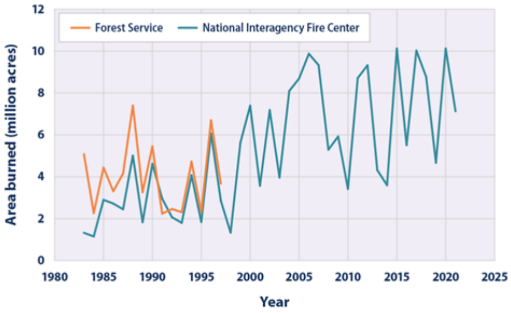
1 Introduction

1.1 Our client

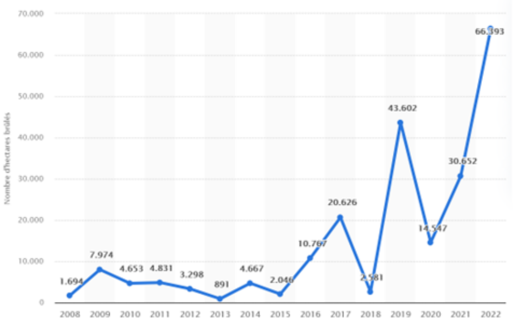
The client of our project is **Frédéric Magoules**, a researcher and lecturer from the Mathematics and Computer Science for Complexity and Systems laboratory (MICS). He previously supervised a group of students who started this project, creating a pipeline for the use of artificial intelligence for detecting wildfires with the use of sound analysis. Frédéric Magoules is a researcher at CentraleSupélec working on projects ranging from parallel computation to IA. At the MICS laboratory, researchers are involved in projects in bio-mathematics, quantitative finance and decision modeling.

1.2 Problem, and expected solution

As a terrible consequence of climate change, forest fires have been on the rise in the past years. Across all continents, scientists are striving to fight these fires through all means.



(a) Forest area burned in the US between 1983 and 2021 (US Environmental Protection Agency)



(b) Forest area burned in France between 2008 and 2022 (Statistica 2023)

Classic detection techniques are usually based on the analysis of image signals. Using either cameras in the wild or satellite imagery, algorithms are able to identify markers of a forest fire and classify the signals in consequence. However, recent developments have included techniques to detect the presence of forest fire through the use of sound signals [1]. By implementing AI in these detection techniques scientists hope to develop a cheaper, more efficient method of protecting the world's forests. [2]

In this project, we will thus try to **detect wildfires by using sound signals**. To do so, we will implement two complementary modules:

- **Feature Extraction**: extract interesting data from the raw forest sounds, reducing the size of the file in the meantime
- **Classifiers**: classify the audio signals using the features obtained with the previous step. From there, select the important features and detect whether the sound is or is not a forest fire.

Figure 1: Forest fire in the Var region - 2022

We thus have the following objectives for our project:

- **State of the art**: Identify the general state of research on sound-based wildfire detection, or more generally on environmental sound detection.
- **Sound databases**: Find and merge sound databases which contains forest fires
- **Implementation**: Implement features extraction techniques and various classifiers that have not been implemented our client, building on their previous code.
- **Evaluation**: Compare the success rate of different methods, on databases containing forest fire samples.

Below, a diagram presents different methods identified by the previous group of students. Some of them were implemented in the previous code.

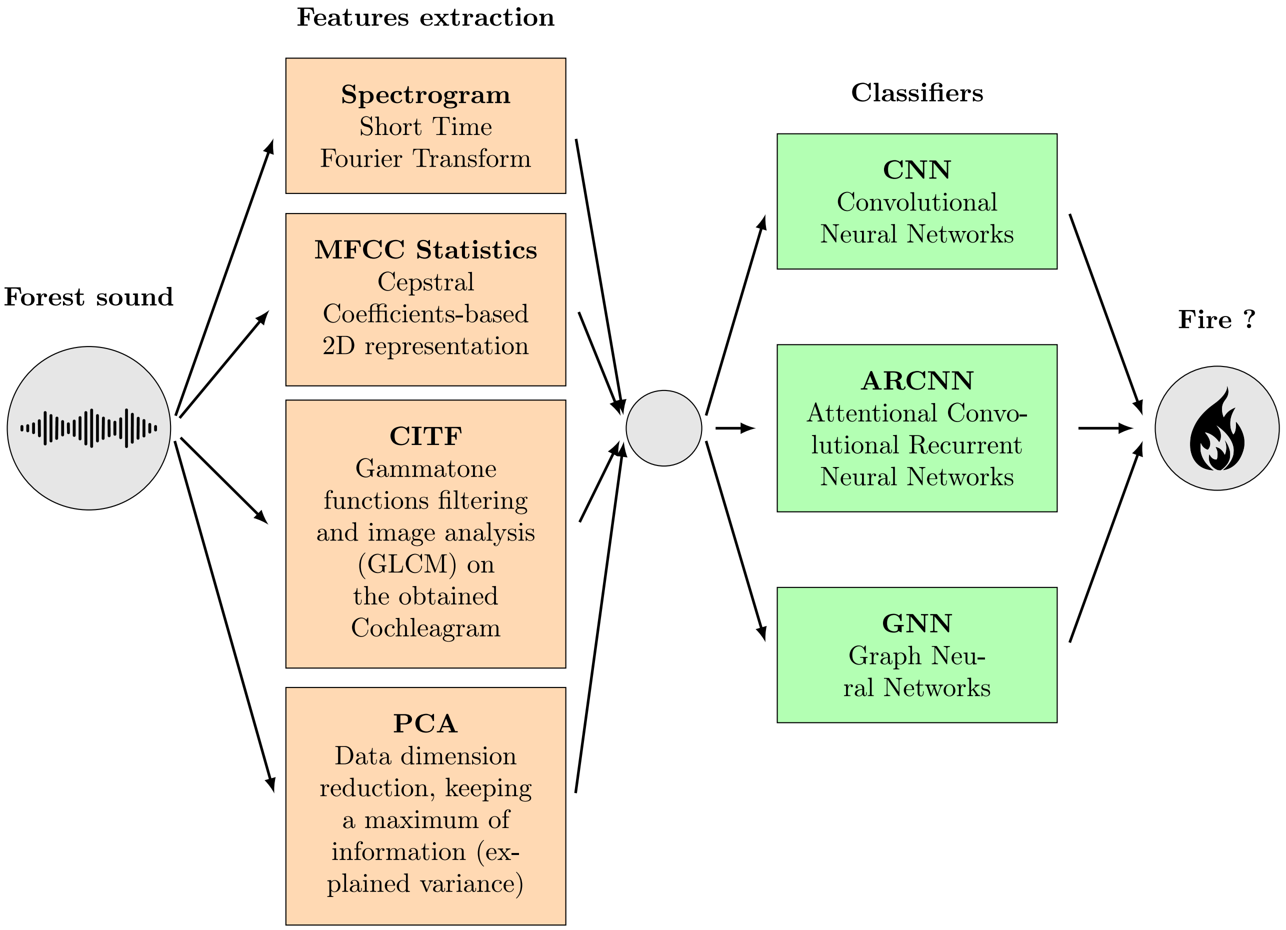

Features extraction
Spectrogram
Short Time Fourier Transform
MFCC Statistics
Cepstral Coefficients-based 2D representation
CITF
Gammatone functions filtering and image analysis (GLCM) on the obtained Cochleagram
PCA
Data dimension reduction, keeping a maximum of information (explained variance)
Forest sound
Classifiers
CNN
Convolutional Neural Networks
ARCNN
Attentional Convolutional Recurrent Neural Networks
GNN
Graph Neural Networks
Fire ?

Figure 2: Functional diagram of the client's existing framework

We will explore different techniques and attempt to refine this existing schema. The provided code relied on ESC-50, a 50-class database covering everything from car sounds to forest fires. Constructing a suitable database will thus be one of the guiding lines of our project.

2 State of the art

2.1 Context: Forest surveillance

Wildfires pose severe dangers to ecosystems and communities all around the world. Thus, having effective systems to detect these wildfires would be highly beneficial. More generally, having an **effective surveillance of large forest areas** can help detect not only wildfires, but also illegal logging of trees, poaching, weather changes, footsteps, and more. In our project, we'll focus on **wildfire detection**.

With the advancements in artificial intelligence technologies and in the Internet of Things (IoT), there as been a newfound interest in effective forest surveillance systems [1]. Indeed, various attempts have been made with satellite images [3], video recording [4], sound recordings [5], or even vibration sensors [6]. However, the first two approaches are expensive and require costly sensors [7, 8]. These are generally not relevant to our project: the last approach is for instance used for illegal logging. Finally, **acoustic surveillance** is cheaper and seems to be a more feasible solution: it is the scope of our project.

Several studies were made on forest acoustic surveillance but usually focus on other problems [5, 6] and not on wildfire detection. For the purpose of wildfire detection, most articles used satellite or video surveillance combined with classifiers to identify the presence of such fires.[9, 10, 11, 12].

Researchers usually discuss the acquisition phase of their data (how to record continuously the signal, process it into segments and what parts to keep) and how to deploy the classification model (whether on a remote server or on the installed device itself). Due to the short nature of our project, we will not focus on these issues.

2.2 Environmental Sound Classification (ESC)

Few papers have been published on the topic of sound-based wildfire detection. We will thus firstly look at a broader issue titled **Environmental Sound Classification (ESC)**. As frequent publications address this problem, it will be an excellent starting point for our study of the state of the art.

As a general definition, *Environmental Sound Classification* consists of classifying environmental sounds, ranging from forests to urban and oceanic sound samples. Classification was first achieved using **Machine Learning** approaches, but recent advances in **Deep Learning** have caused increased attention from the research community towards these techniques, thanks to better self-learning capabilities, and generally more precise and portable results [13].

Several surveys explain the general practices for Environmental Sound Classification [14, 15], but seldom focus on forest fires (there is only one survey on forest sound classification which can be found here [1]). We will eventually have to adapt our techniques to this specific problem.

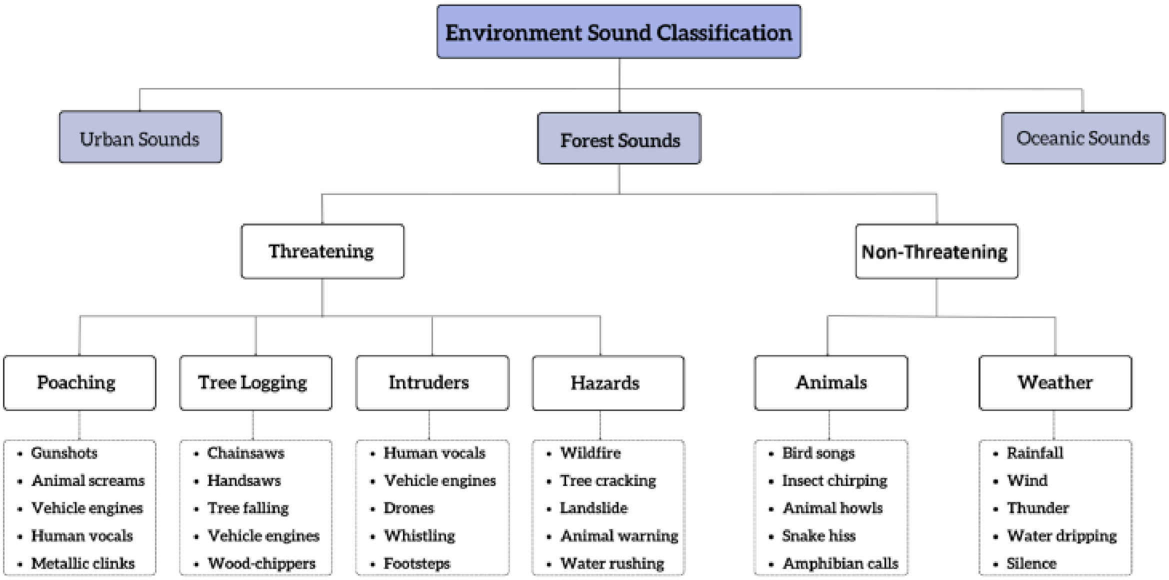


Figure 3: Overview of ESC, with a special focus on forest sounds. *Taken from [1]*

2.3 Sound Pre-processing

In order to classify audio files, one needs to process the raw sound files. Researchers often apply a 3 steps-approach : **audio normalization, data augmentation, and feature extraction**. We will look at these steps in that order.

2.3.1 Audio normalization

Different audio samples are usually recorded at different loudness levels and can contain noise and interference due to the presence, in nature, of various random noisy events. Normalization techniques allow the collection of consistent data that will make for better training data for Machine Learning or Deep Learning models. The table below presents some of these techniques.

Name	**Summary**	**Articles**
Sampling Frequency (SF)	This technique is used to adjust the rate at which audio samples are taken (16 kHz or 44.1 kHz for example).	[16] [17] [18]
Box-cox transform	A statistical method used for transforming data to achieve normality.	[19]
Pre-Emphasis Filters	Audio processing technique to emphasize high-frequency components. Commonly applied in speech and audio signal processing. Boosts the amplitudes of higher frequencies to enhance signal quality.	[20] [21] [22] [23]
Peak Normalization	Adjusting the amplitude of an audio signal to ensure that the highest point (peak) reaches a specified level. Helps prevent distortion and ensures consistent volume levels.	[24] [25]
Root Mean Square	This technique is used to evaluate the acoustic pressure over a short time using a root mean square evaluation	[26] [25]
Bit Depth of Audio	Refers to the number of bits used to represent each sample in digital audio. Higher bit depth allows for greater dynamic range and improved audio fidelity.	[27]
European Broadcast Union Standard	A set of standards defined by the European Broadcasting Union (EBU) for audio signal processing in broadcasting. Addresses issues like loudness normalization to ensure consistent audio levels.	[25]
Log Scaling	Scaling method using logarithmic transformation. Applied to audio for compression or expansion of amplitude ranges. Useful in managing dynamic range and controlling extreme values in audio signals.	[27]

Table 1: Comparison table of normalization methods

The standard is usually to adapt sampling frequency and apply pre-emphasis filters. By changing the **sampling frequency**, usually to a higher sample frequency, it will generate a better accuracy of the original audio file while preserving the nuances and details of the original file. Meanwhile, **Pre-emphasis filters** improve the quality and clarity of audio signals by boosting the higher frequency elements of audio signals.

In our project, we will not delve into audio normalization as the main issue we will face is the lack of data rather than its quality (see the corresponding section below). However, we will take audio normalization in consideration when creating or merging several datasets. At the very least, we will adapt the **Sampling Frequency** for all of our data samples to have the same shape.

2.3.2 Data augmentation

Machine learning models usually require large amounts of data for training, especially for deep learning methods. Unfortunately, most sound classification datasets are quite limited (with the exception of AudioSet, with 2.1 million audio samples). Data augmentation techniques can be used to overcome this issue and artificially grow the database, eventually preventing over-fitting in the training. In projects on ESC, data augmentation is often used to overcome the physical limits of databases[15].

Thus, various signal transformation techniques are used in literature to allow the creation of "new" data files from existing samples to build a larger database: time stretching, pitch shifting, dynamic range compression, adding a random noise [28, 29, 30] ... The common solution for researchers is to generate new data from scratch, usually by using a generative adversarial networks (GAN) [31, 32].

All these approaches can be used in our wildfire detection project to overcome the lack of data. Below are presented a few existing data augmentation techniques:

Name	Summary
Sliding Windowing	Creates dynamic overlapping windows for sound analysis, allowing for efficient trend and pattern detection
Time Stretching	Stretches a sample without modifying the pitch by altering the frequency content
Pitch Shifting	Shifts the pitch without modifying the duration of a sound by altering the frequency content
Spec Augmentation	Involves applying random modifications to spectrograms. Allows the creation of large datasets.
Random Time Delays	Introduces noise in audio samples by applying random modifications. Allows users to produce randomly modified sounds
Signal Speed Scaling	Adjusts the rate of a sound without altering its pitch. It can be applied differently to different parts of a sound.
Generative Adversarial Network	Create new samples from scratch. Class-conditioned synthesis models are trained using an adversarial training strategy by imitating existing data samples.

Table 2: Comparison table of data augmentation methods

2.3.3 Feature extraction

Once the audio samples have been normalized, we can now apply feature extraction techniques to extract relevant information. For feature engineering methods, we have identified the following techniques :

Name	Summary	Articles
Short Fourier Time Transform	Good for analysis of discrepancies in sound samples as it reveals spectro-temporal structures hidden in the phase spectrum.	[19] [33] [34] [35]
Mel Factor Cepstral Coefficients	These coefficients represent the short term power spectrum of a sound. Processing chains includes framing to obtain stationary signals, power spectrum calculation and Mel Filter Bank usage. The Mel Filter Bank was created for speech recognition and creates specific filters. The sound is determined by the type of fire, and different fires have different trendlines. The obtained spectrum of fire shares many similarities with the one of rain.	[19] [33]
Spectrogram	Spectrograms help separate the sounds into different ranges: combustion mostly emits sound in the infra-sound frequency range. However sound depends on what material is being burned and other environmental factors.	[34]

Principal Component Analysis/ Linear discriminant analysis	PCA and LDA are statistical techniques for reducing the dimensionality of a dataset. PCA allows to find the directions of maximum variance in the data and LDA allows to find the projection that best separates the classes.	[19]
Mel-scale Spectrogram	The idea is to combine Mel Factor with a spectrogram to represent the data from Mel-Scale on a easily usable spectrogram.	[33]
Continuous Wavelet Transform and Scalogram	Wavelets are an alternative to the short-term Fourier transform and also enable to surpass the limitations of the classical Fourier transform. In this transformation, the window limiting the STFT is replaced by a family of wavelet functions.	[35] [34]
Fast Fourier Transform	This transformation enables the identification of the frequency content of the signal, but does not enable the temporal localization of events such as jumps or impulses.	[35]
Distribution Wigner-Ville	This transformation enables the analysis of sounds through a different auto-correlation function, then transformed with a a Fourier distribution. It corresponds to the distribution of energy depending on time and frequency	[34] [36]
Zero Crossing Rate	It measures the rate at which a signal changes its sign, passing from positive to negative or vice versa. Zero crossing rate is often employed in speech and audio processing to extract features that can be used for various applications such as speech recognition, speaker identification, and emotion detection. It is quite sensitive to noise.	[33]
Spectral centroid	Consists in reducing the data to a single average, that indicates a central frequency	[19]
Bark/ERB → Cochleagram	The Bark is the standard unit corresponding to one of the 24 critical bands representing the width of human hearing system. Human hearing critical bands are narrow at low frequencies, but become wider at higher frequencies. The equivalent rectangular bandwidth or ERB is a measure used in psychoacoustics, which gives an approximation to the bandwidths of the filters in human hearing, using the unrealistic but convenient simplification of modeling the filters as rectangular band-pass filters, or band-stop filters, like in tailor-made notched music training (TMNMT).	[33]

Table 3: Comparison table of feature engineering methods

After careful consideration, we will implement in our project the most common methods: **Mel and Linear Cepstral Coefficents, Short Time Fourier Transform, Cochleagram, Scalogram, Zero Crossing Rate and Continuous Wavelet transformations**.

2.4 Classifiers

After preprocessing our data, we need to classify it. Sound classification was first made with classical **machine learning** techniques, such as KNN, Decision Trees, Gaussian Mixture Models, or SVM. But with technical advancements, sound classification approaches evolved to **deep learning** methods, such as CNN, RNN, and Transformers (attention mechanisms).

For **machine learning** methods, we have identified the following techniques for sound classification:

Name	**Summary**	**Articles**
SVM	Support Vector Machine. It was a popular supervised machine learning classifier. There are different types of SVM, depending on the kernel for example (linear, polynomial, gaussian...).	[37] [38] [39] [40] [41]

LDA	Based on Bayes' rule. Assumes that the distribution of P(x|C) follows a multivariate normal distribution, and both classes share the same covariance matrix.	[42] [43]
Logistic Regression	Calculates the probability of belonging to a class. According to a predefined threshold, it assigns the observation to that class.	[44]
XGBoost	Builds multiple decision trees and aggregates their predictions to enhance the model's accuracy.	[45] [46]
RandomForest	Combines multiple decision trees using a boosting approach and incorporates regularization techniques to enhance the model's performance.	[47] [48]
KNN	Each audio sample is assigned to the class to which most of the nearest neighbors belong.	[38] [49] [39]
GMM	Probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions	[50]

Table 4: Comparison table of machine learning methods

Most of the articles we found on these commonly used machine learning techniques were published at least 5 years ago. However, most recently published articles use **deep learning** methods. We have identified the following deep learning techniques for sound classification :

Name	**Summary**	**Articles**
CNN	Convolutional Neural Networks : for example, Xception, DenseNet, MobileNet, ResNet, VGG, AlexNet, GoogleNet, Inception ...	[51] [52] [53] [54]
RNN	Recurrent Neural Network: Better for time-bound dependencies. Models include LSTM, GRU	[55] [56] [57] [58] [59]
ACRNN	Auto-Conditioned Recurrent Neural Network : Attention mechanism based convolutional RNN architecture	[60]
Transformer models	Neural network architecture originally designed for natural language processing tasks, but successfully adapted to various other domains, including audio classification. It has shown significant results.	[61] [62] [63]

Table 5: Comparison table of deep learning methods

RNN and CNN models are the most widely used methods in the past few years. CNN seems to spark a higher interest than RNN, in models including VGG, Inception, ResNet. Classic examples of RNN-based models include LSTM, Gated Recurrent Unit (GRU) [1]. Besides, Transformers have recently outperformed CNNs when large datasets are available (almost always AudioSet, which is the only large publicly available dataset for audio classification) [51].

Even though these recent articles show that deep learning methods seems to outperform the classical machine learning methods, neural networks are hindered by their requirement of vast amounts of training data. Besides, most articles try to do a multiclass classification, whereas we only require a binary classification, which might be a simpler task.

Thus, in our project, we will try to implement Machine Learning models in order to see if they can be sufficient for our detection problems. We will at a later time compare these approaches with deep learning methods, for instance with a CNN or RNN.

2.5 Databases

In order to efficiently train and test our models, we need to find or construct databases of sounds related to forests, with ideally an sufficient amount of wildfires samples. We will look for a generally diverse dataset to prevent over-fitting, specialization and reduce the bias.

More precisely, a good dataset must fulfill several requirements [1]:

- **Environmental**: different weather conditions, background noises, forest scenarios (river, mountain, etc.), and different times of the day.
- **Microphone**: different microphones, different sound qualities, different audible points from the ground level
- **Properties of Sound**: different loudness (different distances from the fire), different pitch, and different lengths.

Unfortunately, we only found a single public database dedicated to forest sounds (FSC22 [64]). It is quite a recent database, unfortunately lacking in wildfire sounds.

Therefore, we will focus on the more general case of sound environment databases (ESC). We have identified the following databases for environmental sounds containing wildfires :

Name	**Source**	**Properties**	**Classes**	**Pros**	**Cons**
AudioSet [65]	Youtube	2.1 million samples 1445 fire samples 527 classes 10 s	Wide range of human and animal sounds, musical instruments and genres, and common everyday environmental sounds.	Biggest labeled dataset available	Poor quality for some classes (including fires), and unbalanced dataset
FreeSound [66]	Collaborative platform	500 000 samples	Wide range of sounds	Huge dataset	Poor quality sometimes, Unlabeled
ESC50 [49]	From FreeSound	2000 samples 40 fire samples 50 classes 5 s	Animal sounds, human (non-speech) sounds, urban noises, natural soundscapes (including fire), domestic sounds	High quality, balanced dataset	Unbalanced for our binary classification, with wildfires, small dataset
FSD50K [67]	From FreeSound	51 197 samples 509 fire samples 200 classes	Wide range of categories	High number of samples	Not focused on forest sounds, poor quality sometimes
FSC22 [64]	From FreeSound	2025 samples 75 fire samples 27 classes	Forest sounds, including fires	Database with forest sounds only	Small dataset, unbalanced for wildfire classification

Table 6: Comparison table of environmental sounds databases

Since AudioSet only contains poor quality fire samples, it is not suitable for our wildfire detection (with the exception of pre-training). FreeSound is also not usable because it is unlabeled (only poorly worded tags) and FSD50K varies greatly in sound quality from a sample to the next.
We will focus on **ESC50**, due to its quality and its 2015 publishing date. As it is relatively old, many articles discuss its use [68, 69, 70, 71, 72], and on **FSC22**, which is a quite recent database (2023), due to its quality and abundance of forest sounds.
Because of the lack of fire sounds, we have recovered fire sounds from various other sources :

- Videos from YouTube: many sounds are available when searching on YouTube. However, they are often protected by copyright, and do not give their sources. But it is still possible to construct a self-made private dataset from several YouTube videos.
- A dataset from Kaggle : there is no information about the source of the samples. But it is likely that it was made from videos like this one, which are samples made from scratch by Michael Ghelfi.
- Fires from the Yellowstone National Park, which can be found here.

3 Realization

3.1 Task execution

As a reminder, our goal is to produce a comparative study of different feature extraction techniques and classifiers to detect forest fires.

Our project started at the end of September 2023, and lasted 4 months (end in January 2024). We spent the first month writing the State of the Art of Environmental Sound Classification (ESC) and looking for publications on wildfire detection using sounds.

After that, we spent another month trying to implement and compare the feature extraction techniques and classifiers identified in the state of the art, building on the provided code given by Prof Magoules, and on the dataset ESC50. At the same time, we were finishing our research on publications, in case some important methods or databases were omitted or recently published.

The last two months were spent constructing our final repository, and building new datasets.

The pipeline of our project was difficult to predict given the fact that we were discovering new techniques / challenges every week. However, each person was assigned to a task: most of the time, Karina and Noé focused on feature extraction methods, while Jad, Elias and Erwin dealt with classifiers. Moreover, Elias also focused on the creation of the new dataset, Erwin on the deep learning methods (CNN), Noé and Jad on the object oriented code, and Karina on data augmentation techniques.

3.2 Technical elements

Our project has been coded with an object oriented approach and tailored for usage in a pipeline optimized for Python's **Torch** module. The entirety of our project is available in a GitLab repository. A step-by-step google Colab notebook is also available here.
The diagram below illustrates the way our project is organized.

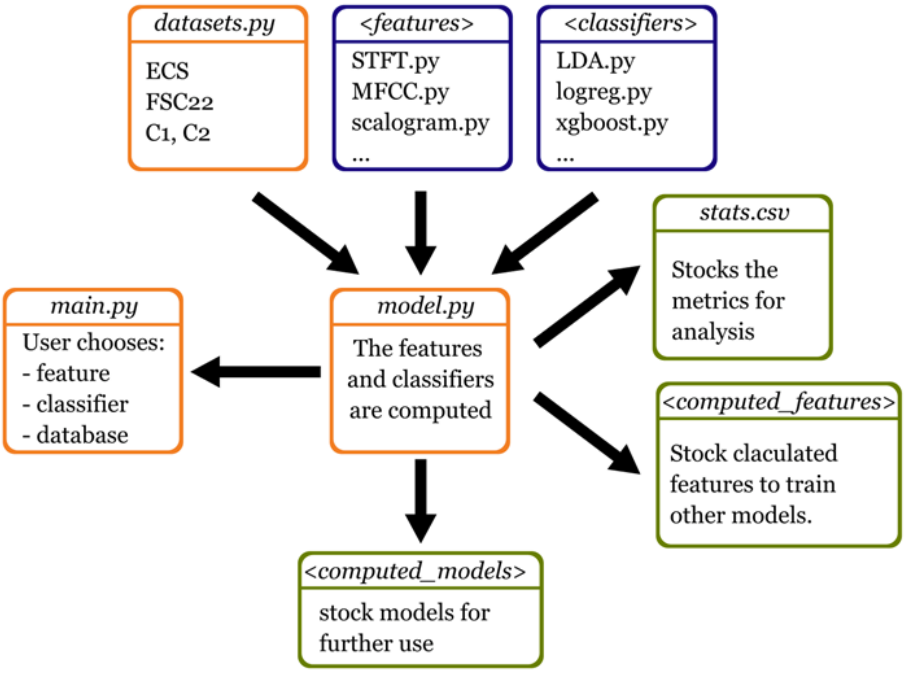


Figure 4: Organizing our work

In orange are represented the main scripts in the home directory of the repository. The *main.py* file allows easy training, testing and usage of our models. It relies on *model.py*'s functions for training and loading and on the classes of classifiers and feature extraction methods, respectively stored in $./classifiers$ and $./features$. When called, the training function search for, and, if a specific combination has not been computed yet, store their computed features, in the $./computed\_features$ directory and computed models in $./computed\_models$. When in training and testing, the script also automatically stores the metrics in *stats.csv*, allowing for a convenient comparison later on. Computed models can easily be applied to a folder containing audio files from the *main.py* file. All the datasets, features and classifiers are implemented as classes, and instances are created for training and testing.

Figure 5 shows a summary of the methods implemented. In green are the feature extraction methods implemented and in orange are the classifier methods implemented. The objective is therefore to know from a sound acquisition whether or not a fire is present (binary classification).

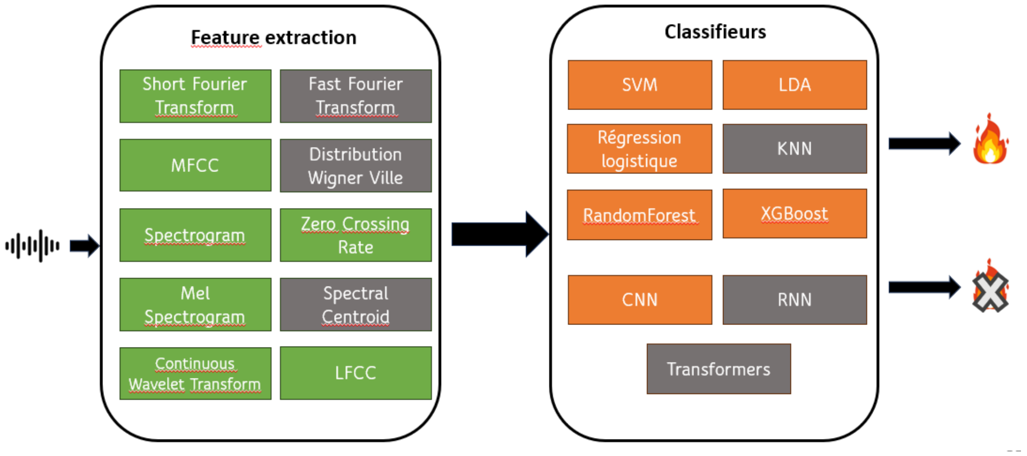


Figure 5: Methods implemented

3.3 New databases

Before implementing feature extraction and classifier methods, it is essential to ensure that relevant databases are available to properly train our classifiers. However, the databases presented in the state of the art have several problems :

- Very unbalanced databases (99% not wildfires in ESC50)
- Small database (only 2000 sounds in ESC50)

To solve these problems, we have merged and created several new databases: $CustomDataset1$ and $CustomDataset2$ and $Youtubetest$. These are created by adding new forest fire sounds to ESC50 and FSC22, coming from a Kaggle database aptly named $KaggleFire$ and from Yellowstone National Park. In addition, a $Youtubetest$ database has been created. The latter comes from a YouTube video which has been split in 5-second extracts and exported as a new database. The result is a database of over 2600 sounds, including 1000 forest fire sounds, a much more balanced database than the previous available ones. This database will be used to validate and test the model's performance. To summarize, we have created the following databases :

- $CustomDataset1$ (dl) : ESC50 + Yellowstone + Kaggle $\Longrightarrow$ Total of 2282 Sounds (322 fire)
- $CustomDataset2$ (dl): FSC22 + Yellowstone + Kaggle $\Longrightarrow$ Total of 2388 Sounds (438 fire)
- $YoutubeTest$ (dl): Youtube 2600 sounds $\Longrightarrow$ Total of 2600 Sounds (1000 fire)

We will train our different models mainly on the $CustomDataset2$ database and we will present our results on this database. The $Youtubetest$ database will be used to validate the tests and performance of the methods implemented. Please refer to the code on GitLab for more details about the *data* classes.

3.4 Feature extraction implemented

For our Feature extraction scripts, we used a simple approach based on two main functions in the classes :

1. $\_\_init\_\_()$ with all the parameters
2. $forward()$ which allows to send the calculated features to the model.

This approach is taken from the $Torch$ module, allowing us to easily unify our different components for ease of use.

3.4.1 MFCC

The Mel Frequency Cepstral Coefficients have been implemented using two different techniques. A first file implements the $MFCC$ class from the $Torch$ module, while a second one uses the $MFCC$ from $Librosa$. Parameters include setting the number of Mel coefficients, the window length, sample rate and the hop size.
As Librosa takes only $np.array$ as an argument, some workaround is require to allow for $torch.tensor$ to be parsed through and from Librosa's functions. Across is a rendered MFCC map from a sample taken from $ECS50$.
MFCC have the combined advantages of being reliable in our models and quick to compute. This allowed us to quickly choose good parameter's for our models.

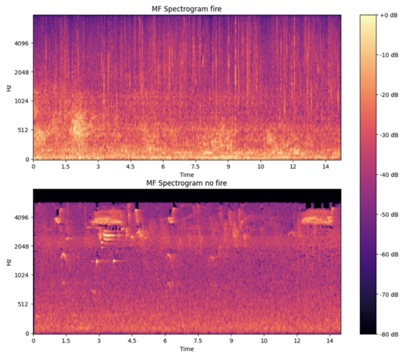


Figure 6: MFCC for sounds from ESC

3.4.2 STFT

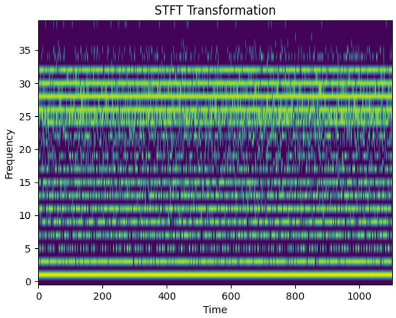


Figure 7: STFT for a ESC label

The Short Time Fourier Transform was implemented using $Torch$'s dedicated function. An STFT corresponds to the following formula:

$$X(\tau, \omega) = \int_{-\infty}^{\infty} x(t) \cdot w(t - \tau) \cdot e^{-j\omega t} \, dt$$

where $x(t)$ is the signal, $w(t - \tau)$ is the window function, τ is the time shift, ω is the frequency, and $X(\tau, \omega)$ is the STFT. We also have implemented a **Mel STFT** which applies a Mel filter to a standard STFT.

3.4.3 LFCC

The Linear Frequency Cesptral Coefficients is a variation of the MFCC in which the filter bank is chosen on a linear scale instead of a Mel scale. It thus corresponds more truly the linear aspect of sound. The implementation is done using $torchaudio$.

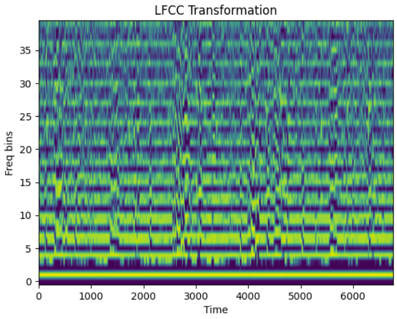


Figure 8: LFCC for a sample from ESC

3.4.4 Scalogram

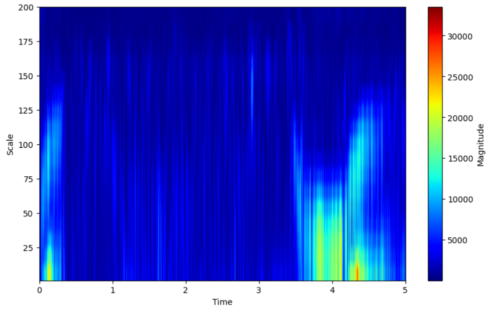


Figure 9: Scalogram for a sample from ESC

The Scalogram has been implemented using the *pywt* module. Again, this required the transformation from *np.array* to and back from *torch.tensor*.
The available arguments include using different Wavelets, sampling period and convolution method.
To reduce the computation time, the Scalogram module extracts a half of the sound to allow for efficient computation. A Scalogram is computed using the formula:

$$S(a,b) = \left| \int_{-\infty}^{\infty} x(t) \cdot \psi^* \left(\frac{t-b}{a} \right) \, dt \right|^2$$

where:

$x(t)$ is the input signal.
$\psi^*(t)$ is the complex conjugate of the analyzing wavelet.
a is the scale parameter, controlling the width of the wavelet.
b is the translation parameter, controlling the position of the wavelet.

3.4.5 Cochleagram

A Cochleagram is supposed to imitate the human Cochlea. Our scripts implements the specifically created Cochleagram module. Computation time is quite high, so this will not be one of our top feature extraction models in our project.

3.4.6 Zero Crossing Rate

The ZCR is a simple method, which represents the rate at which the signal changes its sign (crosses the zero amplitude line) within a given time frame. In simpler terms, it calculates how often the waveform of an audio signal crosses the horizontal zero axis.

3.4.7 PCA

PCA, or Principal Component Analysis, allows us to reduce the size of our inputs. It fits in to a larger effort of reducing the size of the inputs of our models. Indeed, computed features like MFCC are in the forms of matrices, to the size of $128x216$ elements. We can reduce this by choosing an ideal feature subset using PCA. For example, we used PCA to reduce the dimensionality of our data after doing the flattening, between feature extraction methods and classifiers. It was implemented using *sklearn*'s functionalities.

3.5 Transition Features ⇒ Classifiers

Once the various feature extraction methods have been implemented, we now need to provide this as an input to the classifiers. However, two problem arise. The features obtained using the different methods are not homogeneous and cannot be combined for machine learning methods. Moreover, the size of computed features causes our methods to be quite slow. To solve these problem, we need to implement several solutions to standardize the features. The following figure shows the 3 solutions proposed.

- The first solution is to do nothing and keep the output tables from the feature extraction methods. The advantage of this solution is that the data remains untouched, which saves time, but the machine learning methods cannot be used. Only deep learning methods such as CNN will be able to take this data as input.
- The Mean and Mean Var methods are fairly similar and involve calculating the mean or the mean and the variance for each column. The output is therefore a 1-dimensional vector that is compatible with the input of the machine learning methods.
- The Flatten method is undoubtedly the simplest. It simply involves concatenating all the data in the table into a single line. The vector is therefore much larger than that obtained with the Mean or Mean Var methods.

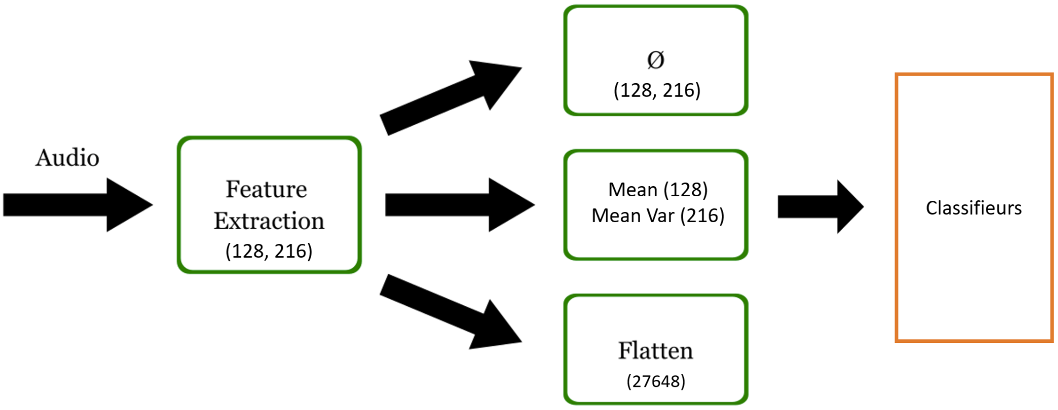


Figure 10: Transition Features ⇒ Classifiers

3.6 Classifiers

In this section, we will present the different classifier methods we have implemented. Our models interact with the rest of the project using the following three functions from the *models.py* file:

- **Train function** : used to train the model from a given dataset
- **Test function** : able to test and outputs metrics from a dataset
- **Evaluate function** : allows the prediction of a model on a new folder of audio files

In the following section, we'll present each implemented classifier, and a sample of its results on the **test set**.

3.6.1 LDA

LDA, or Latent Dirichlet Allocation, is a statistical modelling method often used to discover hidden themes in a data set. By applying LDA to forest fire detection from audio acquisitions, 'latent themes' are extracted from the sound features. The idea is that each audio recording is a combination of these themes, making it possible to identify specific sound patterns associated with forest fires.

Model	**Feature**	**Feature Transf**	**Dataset**	**Accuracy**	**F1-Score**	**Kappa (Cohen)**	**AUC**
LDA	MFCC	mean_var	CustomDB2	0.975	0.927	0.912	0.99

3.6.2 Logistic regression

Logistic regression is a supervised learning method used for classification. In the context of forest fire detection based on audio acquisitions, logistic regression analyses the extracted features, estimates the probability to belongs to a particular class by applying a logistic function and create an optimal decision boundary between the different classes, making it possible to classify sound recordings according to their probability of being linked to a forest fire.

Model	**Feature**	**Feature Transf**	**Dataset**	**Accuracy**	**F1-Score**	**Kappa (Cohen)**	**AUC**
logisticregression	MFCC	mean_var	CustomDB2	0.955	0.884	0.856	0.988

3.6.3 SVM

SVM, or Support Vector Machine, is a supervised learning method used for classification and regression. In the context of forest fire detection based on audio acquisitions, SVM analyses the extracted features to create an optimal decision boundary between the different classes, making it possible to classify sound recordings according to their probability of being linked to a forest fire.

Model	**Feature**	**Feature Transf**	**Dataset**	**Accuracy**	**F1-Score**	**Kappa (Cohen)**	**AUC**
LDA	MFCC	mean_var	CustomDB2	0.975	0.927	0.912	0.99

3.6.4 Random Forest

Random Forest is an ensemble learning method that combines several decision trees to improve the robustness and accuracy of predictions. Applied to forest fire detection from audio acquisitions, Random Forest analyses the extracted features to form a diverse set of decision trees. Each tree contributes to the classification, and the forest makes a consolidated decision based on their individual results.

Model	**Feature**	**Feature Transf**	**Dataset**	**Accuracy**	**F1-Score**	**Kappa (Cohen)**	**AUC**
randomForest	spectrogram	mean_var	CustomDB2	0.986	0.961	0.953	0.998

3.6.5 XGBoost

XGBoost, or eXtreme Gradient Boosting, is an ensemble learning method based on boosted decision trees. In the context of forest fire detection based on audio acquisitions, XGBoost analyses the extracted features to build a set of sequential decision models. Each model adjusts to the errors of the previous one, thereby increasing the overall accuracy of the prediction.

Model	**Feature**	**Feature Transf**	**Dataset**	**Accuracy**	**F1-Score**	**Kappa (Cohen)**	**AUC**
xgboost	MFCC	mean_var	CustomDB2	0.986	0.961	0.952	0.999

3.6.6 CNN

The CNN (Convolutional Neural Network) was first made for image classification. However, since most of our feature extraction methods produce 2D arrays, it is possible to easily use it for audio classification. Our implementation was based on the article [51], which provides a code for an audio classifier using MobileNet, pretrained on the AudioSet database. Below are the results with further training on CustomDB2.

Model	**Feature**	**Feature Transf**	**Dataset**	**Accuracy**	**F1-Score**	**Kappa (Cohen)**	**AUC**
CNN	MelSpectrogram	None	CustomDB2	0.986	0.961	0.953	0.999

3.6.7 Results Table

Below is the full results of the combined performances of the different feature extraction methods and classifiers.
We have chosen 4 metrics to evaluate these methods: AUC, Accuracy, Cohen's Kappa and F1-score. Metrics are calculated using our **CustomDataset2**, with the **test_set**. We have used the Mean Var transformation method to keep a reasonable amount of features.

Model	**Feature**	**Feature Transf**	**Dataset**	**Accuracy**	**F1-Score**	**Kappa (Cohen)**	**AUC**
LDA	spectrogram	mean_var	CustomDB2	0.83	0.47	0.37	0.849
LDA	mel_spectrogram	mean_var	CustomDB2	0.858	0.549	0.466	0.852
LDA	MFCC	mean_var	CustomDB2	0.975	0.927	0.912	0.99
LDA	LFCC	mean_var	CustomDB2	0.925	0.78	0.735	0.953
LDA	MFCC_lib	mean_var	CustomDB2	0.941	0.824	0.789	0.976
LDA	STFT	mean_var	CustomDB2	0.891	0.636	0.575	0.929
LDA	scalogram	mean_var	CustomDB2	0.835	0.289	0.228	0.792
LDA	zero_crossing	mean_var	CustomDB2	0.818	0.0	0.0	0.53
logisticregression	spectrogram	mean_var	CustomDB2	0.796	0.637	0.519	0.955
logisticregression	mel_spectrogram	mean_var	CustomDB2	0.715	0.56	0.405	0.94
logisticregression	MFCC	mean_var	CustomDB2	0.955	0.884	0.856	0.988
logisticregression	LFCC	mean_var	CustomDB2	0.899	0.757	0.695	0.949

logisticregression	MFCC_lib	mean_var	CustomDB2	0.919	0.808	0.758	0.966
logisticregression	STFT	mean_var	CustomDB2	0.891	0.731	0.664	0.956
logisticregression	zero_crossing	mean_var	CustomDB2	0.425	0.309	0.035	0.53
logisticregression	scalogram	mean_var	CustomDB2	0.894	0.725	0.659	0.948
randomForest	spectrogram	mean_var	CustomDB2	0.986	0.961	0.953	0.998
randomForest	mel_spectrogram	mean_var	CustomDB2	0.975	0.931	0.916	0.996
randomForest	MFCC	mean_var	CustomDB2	0.975	0.926	0.911	0.994
randomForest	LFCC	mean_var	CustomDB2	0.975	0.928	0.913	0.994
randomForest	MFCC_lib	mean_var	CustomDB2	0.98	0.946	0.934	0.996
randomForest	STFT	mean_var	CustomDB2	0.969	0.912	0.893	0.99
randomForest	zero_crossing	mean_var	CustomDB2	0.869	0.68	0.599	0.926
randomForest	scalogram	mean_var	CustomDB2	0.975	0.93	0.915	0.995
svm	spectrogram	mean_var	CustomDB2	0.251	0.327	0.033	
svm	mel_spectrogram	mean_var	CustomDB2	0.279	0.335	0.047	
svm	MFCC	mean_var	CustomDB2	0.665	0.483	0.3	
svm	LFCC	mean_var	CustomDB2	0.793	0.58	0.455	
svm	MFCC_lib	mean_var	CustomDB2	0.662	0.476	0.291	
svm	STFT	mean_var	CustomDB2	0.86	0.653	0.566	
svm	scalogram	mean_var	CustomDB2	0.911	0.746	0.692	
svm	zero_crossing	mean_var	CustomDB2	0.62	0.352	0.136	
xgboost	spectrogram	mean_var	CustomDB2	0.983	0.954	0.944	0.999
xgboost	mel_spectrogram	mean_var	CustomDB2	0.983	0.955	0.944	0.994
xgboost	MFCC	mean_var	CustomDB2	0.986	0.961	0.952	0.999
xgboost	LFCC	mean_var	CustomDB2	0.992	0.977	0.972	0.998
xgboost	MFCC_lib	mean_var	CustomDB2	0.983	0.955	0.944	0.999
xgboost	STFT	mean_var	CustomDB2	0.972	0.921	0.904	0.996
xgboost	zero_crossing	mean_var	CustomDB2	0.905	0.761	0.702	0.936
xgboost	scalogram	mean_var	CustomDB2	0.983	0.954	0.944	0.997
MobileNet (CNN)	MelScalogram		CustomDB2	0.986	0.961	0.953	0.999

Table 7: Result table on CustomDataset2 - test set

These comparisons allow us to conclude that **MFCC and MelSpectrogram** seem to be the most effective feature extraction methods. The Machine Leaning methods, applied to these feature extraction techniques, and the CNN, are all relatively good. For example, on the next page are the ROC curve and the correlation matrix for a RandomForest model using MFCC:

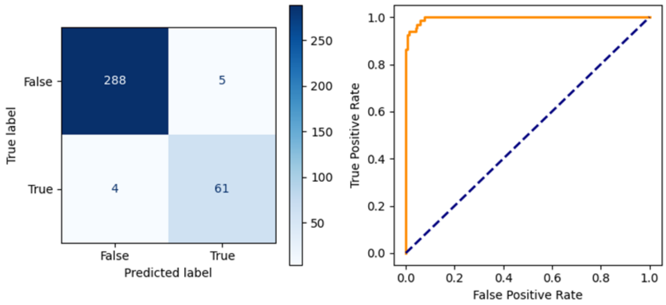


Figure 11: Results for a RandomForest on the test set of CustomDataset2 with MFCC

However, there is a risk that all the sounds of fire in CustomDataset2 are very similar and not representative of reality, thus leading to an over-fitting. Let's test the performance of this RandomForest on our dataset YoutubeTest:

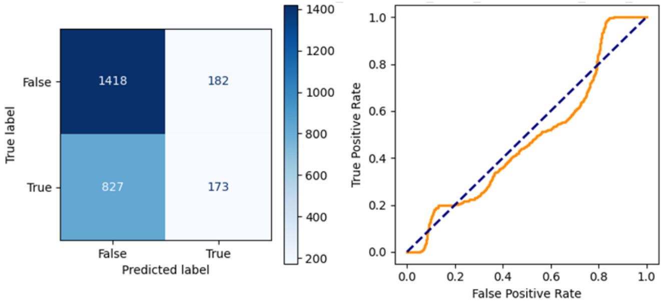


Figure 12: Results for a RandomForest on YoutubeTest

We see that unfortunately, our RandomForest has really poor results when applied to different datasets. This unfortunate conclusion is the same for all our other machine learning methods. Thus, our previous results which seemed to be perfect are hiding a dataset bias. When compared with a random classifier, the RandomForest model is not performing much better on YoutubeTest.
In comparison, the MobileNet CNN provides much-needed positive results. Here are its results for the YoutubeTest database:

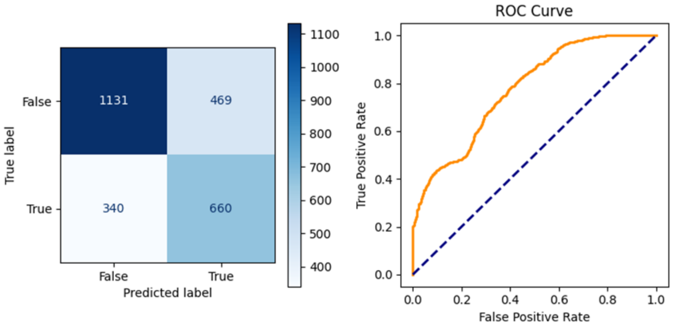


Figure 13: Results for MobileNet on YoutubeTest

Here, we can see that even if our CustomDataset2 might not be representative of the reality, and despite MobileNet being trained on it, it does indeed perform much better than a random classifier (AUC=0.778).

Thus, CNN and more generally deep learning methods seems to have a better generalization potential than machine learning methods.

4 Conclusion and Perspective

4.1 Summary

In this project, we wrote a State Of The Art of techniques of environmental sound classifications, specifically on feature extraction methods and classifiers, with a focus on machine learning methods. We also conducted a review of existing datasets on wildfires and forest sounds, before being faced with the inevitable problem of a lack of data. Thus, we constructed new datasets enriched with wildfires samples. We implemented various feature extraction methods and machine learning classifiers. We have also implemented a deep learning method (CNN). These are easily reusable by our client for the classification of new audio folders.

4.2 What we learned

This project allowed us to learn how to conduct a project with sound samples, how to write a whole project with an object oriented programming. More fundamentally, the project gave us an opportunity to discover many machine learning techniques, and to use a CNN for sound classification. It also helped us to work as a team.

From another point of view, this project has greatly helped us to understand the impact and usefulness of artificial intelligence in the resolution of a real-world problem. We also had the added value of developing our skills in working with a client and as a group.

4.3 Added value to the client

This project has provided great added value to the client. The work of the bibliographic review will allow future students to quickly grasp a global vision of the problem at hand. In addition, new and more complete databases have been created, meaning the client and the students who will work on this project will have a strong starting point to test their models on, with a more coherent dataset than those previously available. Finally, we were able to implement 9 feature extraction methods and 6 classifier methods. This will allow the client to have a good overview of the different methods capable of solving the problem and their limitations, by comparing them using our metrics, but apply them to detect the presence of wildfires. We believe this to be a major step forward in this iterative project.

4.4 Future work

We noticed that neural networks yielded better results on the YouTube database than traditional machine learning methods: more generally, deep learning methods seems to be a promising approach for wildfire detection. Therefore, it would be an avenue worth exploring. Other **deep learning methods (CNNs, RNNs, Transformers)**, using different pre-trained models, could be easily **fine-tuned** to suit our detection problem. Finally, there are evidently many unexplored methods for feature extraction and machine learning that could allows for better detection. Implementing them could move this project forward significantly.

Most importantly, to train and test these methods more effectively, it is crucial to focus on the most significant improvement, which is to build a **larger and more relevant database**. This database should include sounds of various qualities, such as noisy sounds, sounds of different intensities or sounds with background elements like rain or birds. This would help us fight over-fitting. The ideal solution would be to record wildfires sounds, or manually selecting available sounds on YouTube or Freesound. But one should also explore **data augmentation techniques** to create new data and overcome the lack of data.

For future work on this subject, we would thus recommend the following steps:

- Construct or find better databases
- Explore data augmentation techniques
- Explore deep learning further, including by implementing improved CNN, RNN and Transformers methods
- Fine-tuning the models

As a more general conclusion of the project, it is not yet clear whether installing sensors in the forest is a viable solution for wildfire detection We will need to take into account the transmission of audio information without delay, the cost of the equipment and its security. What's more, the sensors must enable us to locate the source of the sound precisely.

References

[1] Dulani Meedeniya, Isuru Ariyarathne, Meelan Bandara, Roshinie Jayasundara, and Charith Perera. A survey on deep learning based forest environment sound classification at the edge. *ACM Comput. Surv.*, 56(3), oct 2023.

[2] TF1. Feux de forêt : comment l'intelligence artificielle aide à les détecter plus tôt. lien. Dernier accès le 10 décembre 2023.

[3] Nina Sofia Wyniawskyj, Milena Napiorkowska, David Petit, Pritimoy Podder, and Paula Marti. Forest monitoring in guatemala using satellite imagery and deep learning. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 6598–6601, 2019.

[4] Osman Günay, Kasım Taşdemir, B. Uğur Töreyin, and A. Enis Çetin. Video based wildfire detection at night. *Fire Safety Journal*, 44(6):860–868, 2009.

[5] Elena Olteanu, Victor Suciu, Svetlana Segarceanu, Ioana Petre, and Andrei Scheianu. Forest monitoring system through sound recognition. In *2018 International Conference on Communications (COMM)*, pages 75–80, 2018.

[6] Dirga Chandra Prasetyo, Giva Andriana Mutiara, and Rini Handayani. Chainsaw sound and vibration detector system for illegal logging. In *2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, pages 93–98, 2018.

[7] Anupriya Prasad and Pradeep Chawda. Power management factors and techniques for iot design devices. In *2018 19th International Symposium on Quality Electronic Design (ISQED)*, pages 364–369, 2018.

[8] Rajesh Singh, Anita Gehlot, Shaik Vaseem Akram, Amit Kumar Thakur, Dharam Buddhi, and Prabin Kumar Das. Forest 4.0: Digitalization of forest using the internet of things (iot). *Journal of King Saud University - Computer and Information Sciences*, 34(8, Part B):5587–5601, 2022.

[9] Ata Akbari Asanjan, Milad Memarzadeh, P. Aaron Lott, Thomas Templin, and Eleanor Rieffel. Quantum-compatible variational segmentation for image-to-image wildfire detection using satellite data. In *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, pages 4919–4922, 2022.

[10] Bilge Memis, Bengisu Kaplan, Boğac Karabulut, and Alim Rustem Aslan. Early detection of wildfires in the european area with nano-satellite constellations. In *2023 10th International Conference on Recent Advances in Air and Space Technologies (RAST)*, pages 1–5, 2023.

[11] Ahmed S. Mahdi and Sawsen A. Mahmood. Analysis of deep learning methods for early wildfire detection systems: Review. In *2022 5th International Conference on Engineering Technology and its Applications (IICETA)*, pages 271–276, 2022.

[12] Steven G. Xu, Seunghyun Kong, and Zohreh Asgharzadeh. Wildfire detection using streaming satellite imagery. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 2899–2902, 2021.

[13] Nitin Kumar Chauhan and Krishna Singh. A review on conventional machine learning vs deep learning. In *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, pages 347–352, 2018.

[14] Olusola O. Abayomi-Alli, Robertas Damaševičius, Atika Qazi, Mariam Adedoyin-Olowe, and Sanjay Misra. Data augmentation and deep learning methods in sound classification: A systematic review. *Electronics*, 11(22), 2022.

[15] Jakob Abeßer. A review of deep learning based methods for acoustic scene classification. *Applied Sciences*, 10(6), 2020.

[16] Riccardo Zambon Alessandro Andreadis, Giovanni Giambene. Monitoring illegal tree cutting through ultra-low-power smart iot devices. *Sensors 2021, 21(22), 7593*, 2021.

[17] Steven Wyatt Evan Martino David Elliott, Carlos E. Otero. Tiny transformers for environmental sound classification at the edge. *arXiv*, 2021.

[18] Alessandro Lameiras Koerich Sajjad Abdoli, Patrick Cardinal. End-to-end environmental sound classification using a 1d convolutional neural network. *arXiv*, 2019.

[19] Hugues Vinet, Gérard Assayag, Juan José Burred, Grégoire Carpentier, Nicolas Misdariis, Geoffroy Peeters, Axel Roebel, Norbert Schnell, Diemo Schwarz, and Damien Tardieu. Sample orchestrator : gestion par le contenu d'échantillons sonores. *Traitement du Signal*, 28(3-4):417–468, 2011.

[20] Hongbo Fei Wei Li Jinghu Yu Zilong Huang, Chen Liu and Yi Cao. Urban sound classification based on 2-order dense convolutional network using dual features. *Applied Acoustics 164*, 2020.

[21] Kalyanaswamy Banuroopa and Shanmuga Priyaa. Mfcc based hybrid fingerprinting method for audio classification through lstm. *International Journal of Nonlinear Analysis and Applications*, 2022.

[22] Sifat Tanvi Kadir Hamim Hassan Iqbal Junaid Mostakim Moin. Imran Mohammed Safwat, Rahman Afia Fahmida. An analysis of audio classification techniques using deep learning architectures. *6th International Conference on Inventive Computation Technologies (ICICT'21). IEEE, Los Alamitos, CA, 805–812.*, 2021.

[23] George Suciu. Svetlana Segarceanu, Elena Olteanu. Forest monitoring using forest sound identification. *Proceedings of the 2020 43rd International Conference on Telecommunications and Signal Processing (TSP'20). IEEE, Los Alamitos, CA, 346–349.*, 2020.

[24] Jaeha Kim Jaehun Kim, Kyoungin Noh and Joon-Hyuk Chang. Sound event detection based on beamformed convolutional neural network using multi-microphones. *Proceedings of the 2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC'18). IEEE, Los Alamitos, CA, 170–173.*, 2018.

[25] Yugyung Lee. Sayed Khushal Shah, Zeenat Tariq. Iot based urban noise monitoring in deep learning using historical reports. *Proceedings of the IEEE International Conference on Big Data*, 2019 IEEE, Los Alamitos, CA, 4179–4184.

[26] Kelefouras V Paraskevas M. Mporas I, Perikos I. Illegal logging detection based on acoustic surveillance of forest. *Applied Sciences*, 2020 10(20):7379.

[27] S. Wich C. Chalmers, P. Fergus and S. N. Longmore. Modelling animal biodiversity using acoustic monitoring and deep learning. *Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN'21). IEEE, Los Alamitos, CA, 1–7.*, 2021.

[28] Jakob Abeßer, Stylianos Ioannis Mimilakis, Robert Gräfe, Hanna M Lukashevich, and IDMT Fraunhofer. Acoustic scene classification by combining autoencoder-based dimensionality reduction and convolutional neural networks. In *DCASE*, pages 7–11, 2017.

[29] Justin Salamon and Juan Pablo Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal processing letters*, 24(3):279–283, 2017.

[30] Jun-Xiang Xu, Tzu-Ching Lin, Tsai-Ching Yu, Tzu-Chiang Tai, and Pao-Chi Chang. Acoustic scene classification using reduced mobilenet architecture. In *2018 IEEE International Symposium on Multimedia (ISM)*, pages 267–270. IEEE, 2018.

[31] Hangting Chen, Zuozhen Liu, Zongming Liu, Pengyuan Zhang, and Yonghong Yan. Integrating the data augmentation scheme with various classifiers for acoustic scene modeling, 2019.

[32] Seongkyu Mun, Suwon Shon, Wooil Kim, David K. Han, and Hanseok Ko. Deep neural network based learning and transferring mid-level audio features for acoustic scene classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 796–800, 2017.

[33] Geoffroy Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Technical report, Icram, 2004.

[34] Vincent Choqueuse El Houssin El Bouchikhi, Mohamed Benbouzid, and Jean Frédéric Charpentier. Etude comparative des techniques de traitement du signal non-stationnaires dédiées au diagnostic des génératrices asynchrones dans les eoliennes offshores et les hydroliennes.

[35] Béatrice PESQUET-POPESCU and Jean-Christophe PESQUET. Ondelettes et applications. *Le traitement du signal et ses applications*, Aug 2001.

[36] Patrick Flandrin and Bernard Escudié. Principe et mise en œuvre de l'analyse temps fréquence par transformation de wigner-ville. 1985.

[37] C.-c. Jay Kuo Maja J. Mataric Selina Chu, Shrikanth Narayanan. Where am i? scene recognition for mobile robots using audio features. *2006 IEEE International Conference on Multimedia and Expo*, 2006.

[38] Vasileios Bountourakis, Lazaros Vrysis, and George Papanikolaou. Machine learning algorithms for environmental sound recognition: Towards soundscape semantics. *AM '15: Proceedings of the Audio Mostly 2015 on Interaction With Sound*, 2015.

[39] N'tcho Assoukpou Jean GNAMELE, Yelakan Berenger OUATTARA, Toka Arsene KOBEA, Geneviève BAUDOIN, and Jean-Marc LAHEURTE. Knn and svm classification for chainsaw sound identification in the forest areas. *International Journal of Advanced Computer Science and Applications*, 10(12), 2019.

[40] Ying Li and Zhibin Wu. Animal sound recognition based on double feature of spectrogram in real environment. In *2015 International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–5, 2015.

[41] Anam Bansal and Naresh Kumar Garg. Environmental sound classification: A descriptive review of the literature. *Intelligent Systems with Applications*, 16:200115, 2022.

[42] G.P. Georgiou. Comparison of the prediction accuracy of machine learning algorithms in crosslinguistic vowel classification. *Sci Rep 13, 15594*, 2023.

[43] Murugiaya Ramashini, Pg Emeroylariffion Abas, Ulmar Grafe, and Liyanage C De Silva. Bird sounds classification using linear discriminant analysis. In *2019 4th International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*, pages 1–6, 2019.

[44] Antonio Robles-Guerrero, Tonatiuh Saucedo-Anaya, Efrén González-Ramírez, and José Ismael De la Rosa-Vargas. Analysis of a multiclass classification problem by lasso logistic regression and singular value decomposition to identify sound patterns in queenless bee colonies. *Computers and Electronics in Agriculture*, 159:69–74, 2019.

[45] Weiyun Jin, Xiao Wang, and Yi Zhan. Environmental sound classification algorithm based on region joint signal analysis feature and boosting ensemble learning. *Electronics*, 11(22), 2022.

[46] Svetlana Segarceanu, Elena Olteanu, and George Suciu. Forest monitoring using forest sound identification. In *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*, pages 346–349, 2020.

[47] Muhammad M. Al-Maathidi and Francis F. Li. Audio content feature selection and classification a random forests and decision tree approach. In *2015 IEEE International Conference on Progress in Informatics and Computing (PIC)*, pages 108–112, 2015.

[48] Md. Rifat Ansari, Sadia Alam Tumpa, Jannat Ara Ferdouse Raya, and Mohammad N. Murshed. Comparison between support vector machine and random forest for audio classification. In *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*, pages 1–4, 2021.

[49] Karol J. Piczak. Esc: Dataset for environmental sound classification. *DOI: http://dx.doi.org/10.1145/2733373.2806390.*, 2015.

[50] Elena Olteanu, Victor Suciu, Svetlana Segarceanu, Ioana Petre, and Andrei Scheianu. Forest monitoring system through sound recognition. In *2018 International Conference on Communications (COMM)*, pages 75–80, 2018.

[51] Florian Schmid, Khaled Koutini, and Gerhard Widmer. Efficient large-scale audio tagging via transformer-to-cnn knowledge distillation. pages 1–5, 06 2023.

[52] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition, 2020.

[53] Yuan Gong, Yu-An Chung, and James Glass. Psla: Improving audio tagging with pretraining, sampling, labeling, and aggregation. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 29:3292–3306, oct 2021.

[54] Sergey Verbitskiy, Vladimir Berikov, and Viacheslav Vyshegorodtsev. Eranns: Efficient residual audio neural networks for audio pattern recognition. *Pattern Recognition Letters*, 161:38–44, 2022.

[55] Pal Abhijit Kumar Dutta Sumit Joy Krishan Das, Ghosh Arka and Chakrabarty Amitabha. Urban sound classification using convolutional neural network and long short term memory based on multiple features. *Proceedings of the 2020 4th International Conference on Intelligent Computing in Data Sciences (ICDS'20). IEEE, Los Alamitos, CA, 1–9.*, 2020.

[56] Haoran Sun Chai Chen, Yuxuan Liu and Moyan Zhou. Audio feature extraction and classification for urban sound. *Retrieved September 11, 2023 from https://github.com/yuxuan3713/ECE-228-Project.*

[57] Jiangtao Hu Lidong Yang and Zhuangzhuang Zhang. Audio scene classification based on gated recurrent unit. *Proceedings of the IEEE International Conference on Signal, Information, and Data Processing. IEEE, Los Alamitos, CA, 1–5.*, 2019.

[58] Yuzhong Wu and Tan Lee. Reducing model complexity for dnn based large-scale audio classification. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'18). IEEE, Los Alamitos, CA, 331–335.*, 2018.

[59] Thi Thuy An Dang and Thi Kieu Tran. Audio scene classification using gated recurrent neural network. *Proceedings of the Conference on Information Technology and Its Applications (CITA'16). IEEE, Los Alamitos, CA, 48–51.*, 2016.

[60] Shunqing Zhang Tianhao Qiao Shan Cao Zhichao Zhang, Shugong Xu. Attention based convolutional recurrent neural network for environmental sound classification. *Neurocomputing Volume 453, 17 September 2021, Pages 896-903*, 2021.

[61] Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. Masked autoencoders that listen, 2023.

[62] Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection, 2022.

[63] Khaled Koutini, Jan Schlüter, Hamid Eghbal-zadeh, and Gerhard Widmer. Efficient training of audio transformers with patchout. In *Interspeech 2022*, interspeech$_2$022.$ISCA, September$2022.

[64] Meelan Bandara, Roshinie Jayasundara, Isuru Ariyarathne, Dulani Meedeniya, and Charith Perera. Forest sound classification dataset: Fsc22. *Sensors*, 23(4), 2023.

[65] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[66] Freesound. `https://annotator.freesound.org/` and `https://labs.freesound.org`. Accessed: 05/02/24.

[67] Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra. Fsd50k: An open dataset of human-labeled sound events, 2022.

[68] Sainath Adapa. Urban sound tagging using convolutional neural networks, 2019.

[69] Alessandro Andreadis, Giovanni Giambene, and Riccardo Zambon. Monitoring illegal tree cutting through ultra-low-power smart iot devices. *Sensors*, 21(22), 2021.

[70] Gianmarco Cerutti, Rahul Prasad, Alessio Brutti, and Elisabetta Farella. Compact recurrent neural networks for acoustic event detection on low-energy low-complexity platforms. *IEEE Journal of Selected Topics in Signal Processing*, 14(4):654–664, May 2020.

[71] David Elliott, Carlos E. Otero, Steven Wyatt, and Evan Martino. Tiny transformers for environmental sound classification at the edge, 2021.

[72] A. Guzhov, F. Raue, J. Hees, and A. Dengel. Esresnet: Environmental sound classification based on visual domain models. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4933–4940, Los Alamitos, CA, USA, jan 2021. IEEE Computer Society.