

**Democratic Republic of Algeria**  
**Ministry of Higher Education and Scientific Research**  
**University of BLIDA 1**

Faculty of Science  
Computer Science Department



**MASTER THESIS**

**Option: Software engineering**

**A Study of Sound Event Detection Techniques  
for Home Activity Monitoring**

By

**Nesrine ZIDANE**

**Sarra ZABAT**

**In front of a jury composed of:**

Ms Guessoum Dalila

President

Ms Bey Fella

Examiner

Ms. YKHLEF Hadjer

Supervisor

Mr. YKHLEF Farid

Supervisor

2019/2020

## **Abstract**

This thesis is dedicated to an experimental study of home acoustic activity monitoring within sound event detection systems. The principal goal is to develop an efficient system for activity classification using a large set of audio activities within DCASE 2018 datasets. We have built a monitoring system by extracting features (Log Mel-Band energies) from time frames of each audio signal. Then, we have trained the extracted features using a deep neural network, namely Convolutional Neural Network (CNNs). Eventually, our study shows that the combination of Log Mel-band Energy features and CNN learning algorithm helps getting a good performance that allows the system to show a strong generalization ability.

**Key words:** Acoustic Activity Monitoring, Sound Event Detection, Feature Extraction, Machine Learning.

## Résumé

Ce mémoire est dédié à une étude expérimentale de la surveillance de l'activité acoustique domestique dans les systèmes de détection d'événements sonores. L'objectif principal est de développer un système efficace pour la classification des activités en utilisant un large ensemble d'activités audio dans les ensembles de données DCASE 2018. Nous avons construit un système de surveillance en extrayant des caractéristiques (Log Mel-Band énergies) à partir des tranches de temps de chaque signal audio. Ensuite, nous avons formé les fonctionnalités extraites à l'aide d'un réseau de neurones profond, à savoir le réseau neuronal convolutif (CNN). Finalement, notre étude montre que la combinaison des fonctionnalités Log Mel-band Energy et de l'algorithme d'apprentissage CNN permet d'obtenir une bonne performance qui permet au système de montrer une forte capacité de généralisation.

**Mots clés:** Surveillance de l'activité acoustique, Détection d'événements sonores, Extraction de caractéristiques, Apprentissage automatique.

## ملخص

هذه الأطروحة مخصصة لدراسة تجريبية لمراقبة النشاط الصوتي في المنزل ضمن أنظمة الكشف عن الأحداث الصوتية. الهدف الرئيسي هو تطوير نظام فعال لتصنيف النشاط باستخدام مجموعة كبيرة من الأنشطة الصوتية ضمن مجموعات بيانات DCASE 2018. لقد قمنا ببناء نظام مراقبة من خلال استخراج الميزات (طاقات Log Mel-Band) من الأطر الزمنية لكل إشارة صوتية. بعد ذلك ، قمنا بتدريب الميزات المستخرجة باستخدام شبكة عصبية عميقة ، وهي الشبكة العصبية التلافيفية (CNNs). في النهاية ، تُظهر دراستنا أن الجمع بين ميزات Log Mel-band Energy وخوارزمية تعلم CNN تساعد في الحصول على أداء جيد يسمح للنظام بإظهار قدرة تعميم قوية.

**الكلمات المفتاحية :** مراقبة النشاط الصوتي ، اكتشاف الأحداث الصوتية ، تقنيات استخراج الميزات الصوتية ، التعلم الآلي.

## **Acknowledgments**

At the first place, we would like to thank our supervisors Ms YKHLEF Hadjer and Mr YKHLEF Farid for their encouragement, advice, and the time they provided for this thesis. We would also thank them for their guidance which made it possible to accomplish this simple work.

Also we address thanks to the reviewers Ms Bey Fella and Ms Guessoum Dalila For the time they spent on reading the thesis, and for their constructive comments which definitely will help for further works.

# Contents

- Introduction ..... 1**
- Chapter 1: Sound representation and Feature engineering .....4**
  - 1.1. Introduction..... 4
  - 1.2. Sound recording .....5
  - 1.3. Multi-channel audio recording .....6
  - 1.4. Time and frequency representation.....6
    - 1.4.1. Fourier transform.....7
  - 1.5. Feature engineering.....8
    - 1.5.1. Framing ..... 8
    - 1.5.2. Windowing .....9
  - 1.6. Spectral features.....10
    - 1.6.1. Log Mel-band energy features.....10
    - 1.6.2. Mel Frequency Cepstral Coefficients (MFCC) .....12
    - 1.6.3. Spectrogram feature ..... 12
  - 1.7. Mean Normalization ..... 13
  - 1.8. Conclusion ..... 14
- Chapter 2: Machine learning for Home Activity Monitoring .....15**
  - 2.1. Introduction..... 15
  - 2.2. Classification ..... 15
  - 2.3. Neural Networks .....17
    - 2.3.1. Convolutional Neural Network .....18
  - 2.4. Evaluation of HAM systems .....22
    - 2.4.1. Cross validation.....22
    - 2.4.2. Measures of Performance .....23
      - 2.4.2.1. Confusion matrix .....23
      - 2.4.2.2. Precision and Recall.....23
      - 2.4.2.3. F-Score.....24
  - 2.5. Challenges within HAM Systems .....24

2.5.1. Overlapping events.....	24
2.5.2. Human attitude .....	24
2.6. Related Work .....	25
2.7. Conclusion .....	26
<b>Chapter 3: Experimental setup and results .....</b>	<b>27</b>
3.1. Introduction.....	27
3.2. Dataset Presentation.....	27
3.3. Tools .....	28
3.4. System Description .....	30
3.5. Experimental Setup.....	31
3.5.1. Log Mel-band Energy Setup .....	31
3.5.2. CNN Setup .....	32
3.6. Experimental results .....	34
3.6.1. Discussion .....	35
3.6.2. Execution Time .....	36
3.7. Conclusion .....	36
<b>Conclusion.....</b>	<b>37</b>
<b>Bibliography .....</b>	<b>39</b>

# List of Figures

Figure 1.1: HAM System mechanism .....	4
Figure 1.2: Frequency in function of time [11] .....	7
Figure 1.3: Signal framing .....	9
Figure 1.4: Hamming window .....	10
Figure 1.5: Log Mel-band Energy extraction process [20] .....	11
Figure 1.6: Filter bank on a male scale [22] .....	12
Figure 1.7: Spectrogram features sample [22] .....	13
Figure 1.8: Mean Normalization [22] .....	14
Figure 2.1: Multi-Layer Neural Network .....	17
Figure 2.2: Architecture of a CNN [30] .....	18
Figure 2.3: Convolution process [32] .....	19
Figure 2.4: 4-Fold Cross Validation [43] .....	22
Figure 3.1: Screenshot of Google Colab notebook .....	29
Figure 3.2: Principal steps for conducting HAM system .....	30
Figure 3.3: Log Mel-band Energy spectrogram features .....	32
Figure 3.4: F1-Scores (%) For each class and Overall F1-Score for CNN100 and CNN500 .....	35



## List of Tables

Table 2.1: HAM related work .....	25
Table 3.1: Frequency of occurrence of 10s segments daily activities in DCASE2018 dataset .....	28
Table 3.2: Log Mel-band energy features Setup .....	31
Table 3.3: CNN Setup .....	33
Table 3.4: F1-Scores (%) .....	34

## List of Acronyms

**DCASE:** Detection and Classification of Acoustic Scenes and Events

**HAM:** Home Activity Monitoring

**SED:** Sound Event Detection

**TFR:** Time–Frequency representation

**DFT:** Discrete Fourier Transform

**FFT:** Fast Fourier Transform

**MFCC:** Mel-frequency Cepstral Coefficients

**STFT:** Short Time Fourier Transform

**DCT:** Discrete Cosine Transform

**SNR:** Signal-to-Noise Ratio

**NNs:** Neural networks

**CNN:** Convolutional neural network

**ReLU:** Rectified Linear Units Activation

**GCNN:** Gated Convolutional Neural Network

**FNN:** Feed Forward Neural Network

**LSTM:** Long Short-Term Memory

**ResNET:** Residual Neural Network

# Introduction

## 1. General context and problematic

**Home activity monitoring (HAM)** systems principally aim to keep awareness of the living environment of a human being to improve quality of life within domestics; these environments are known as **Smart Homes**. For this purpose, considerable researches have been performed by scientists aimed at developing the direction of smart home-based activity monitoring. HAM was primarily introduced to detect the daily routines of elderly people with chronic illnesses ('Alzheimer's disease, dementia, diabetes, cardiovascular disease, osteoarthritis) by tracking their daily activities performed at home and deploying a motion sensors network in different areas of the home. These common conditions, coupled with the naturally occurring progressive decline in physical and cognitive skills of elderly people prevent many from living independently in their domestics. Nowadays, HAM applications have widely been expanded; thus it is used to provide a way of preserving the ability of people to safely remain in their own homes as long as possible by avoiding many potentially dangerous events (fire outbreak, glass breaking, gunshot, theft incidents, external intrusions...) that can be detected at an early stage through the analysis of the home activity data.

Recent advances in communications and computing technologies, along with advances in ambient intelligent technologies, such as **multi-sensors** monitoring systems, have resulted in a rapid emergence of smart living environments equipped with intelligent technologies to follow activities of people. Compared to other modalities, microphone sensors contain highly informative data which can be exploited for multiple purposes since HAM is a subset of Sound Event Detection (SED) field [1,2]. **SED systems are based on recognizing the sound events present in an audio recording** using **Signal Processing** methods then **associating a semantic label to an audio stream** that identifies the concerned events. Tremendous efforts have been put into adapting **sound analysis** techniques for domestic audio recordings (e.g. Cooking, Dishwashing, Watching TV...), then attributing a class label to each activity using algorithms of **Machine learning** that attempt to automatically recognize activities.

**Sound analysis is the main objective of HAM** through the analysis of a big amount of acoustic data. Thus, audio surveillance was motivated by the fact that sounds travel through

obstacles, is not affected by lighting conditions, and capturing sound typically consumes less power [3] which can be advantageous in many ways unlike video recording.

The process of sound analysis consists of two major steps: **Feature Extraction** and **Machine Learning**. First, a **Feature Extraction** method is applied on each frame from the equal length frames of the **multi-channel audio signal** (Find more details in Chapter 1) to extract a feature vector per frame. Various **feature extraction** techniques exist such as: Log Mel-band Energy, Mel frequency Cepstral Coefficients and other low-level **Spectral Descriptors** such as histograms of sound events learned from time-frequency representations. Second, **Machine Learning model** maps the extracted vector of features to the corresponding label. Many classification algorithms have been introduced in the literature: **Neural Networks**, Support Vector Machines and Hidden Markov models [4,5].

Since the acoustic features extracted from a **multi-channel audio recording** are usually composed of multiple overlapping events (**Polyphonic**) which is not as representative as the features of an individual sound event (**Monophonic**), the main problem in real-life monitoring systems is that human attitude at home cannot be predictable in all cases (new activities, Unexpected or rare activities.). Technically, the majority of SED datasets (including HAM datasets) are not very rich in terms of data i.e. they do not contain the sufficient amount of varied data, this fact minimizes the generalization ability of classification approaches to reflect the different characteristics of each activity sound efficiently. A possible solution would be to utilize **data augmentation** techniques to cope with the lack of data issue. Multiple **Data Augmentation** techniques are introduced to improve the generalization ability of their classifiers. These techniques consist of expanding the size of the training set by creating modified (mixed, shuffled) versions of the original audio recordings [6,7].

## 2. Contributions

This work deals with building a **monophonic home activity monitoring system** that is able to analyze sound recordings and indicate whether the activity exists or not in a given set of **monophonic audio recordings**. In this thesis, we focus on building and improving the quality of the system based on the Dataset from DCASE2018. The main objectives of this thesis are summarized as follows:

- First, we have experimented a sound analysis method to build a HAM system based on multi-channel audio recordings. For this purpose, we have tested the combination of Log Mel-band Energy features and Convolutional Neural Network paradigm with varied parameters. This experimentation was conducted using a huge set of data from DCASE 2018.
- Second, we have studied the effect of varying the number of epochs on the generalization ability of CNN in order to address the overfitting.

### **3. Thesis structure**

This thesis is structured in the following way:

- Chapter 1: which is concerned with presenting an overview of acoustic features used for sound representation and feature engineering.
- Chapter 2: In which we provide an overview of classification and **machine learning** concepts that are relevant for this thesis.
- Chapter 3: in which we present the experimental setup and results of our enquiries. We provide a description for the tools and setup that we have deployed for enquiring the dataset, then we present and discuss the results by performance score tables.

# Chapter 1: Sound representation and Feature engineering

## 1.1. Introduction

There is a rising interest in smart environments that improve the quality of life for humans in terms of safety, security, comfort, and home care. In order to have smart functionality, situational awareness is required, which might be obtained by interpreting a multitude of sensing modalities including acoustics. This latter is already used in vocal assistants such as Google Home, Apple HomePod, and Amazon Echo. These devices focus on speech. However, they could be extended to identify domestic activities carried out by humans. Yet, the acoustic models are typically based on single channel and single location recordings. In this study, it is investigated to extend multi-channel acoustic recordings which are beneficial for the purpose of building HAM systems. Usually, the process of sound analysis used for building such systems comprises two primary stages: Feature Extraction or Engineering and Machine Learning. First, feature extraction provides a numerical representation of each frame within a sound recording (signal). Second, machine learning models map the extracted representation of each frame to its label (monitored activity). Figure 1.1 shows a diagram containing the modules that compose the process of HAM.

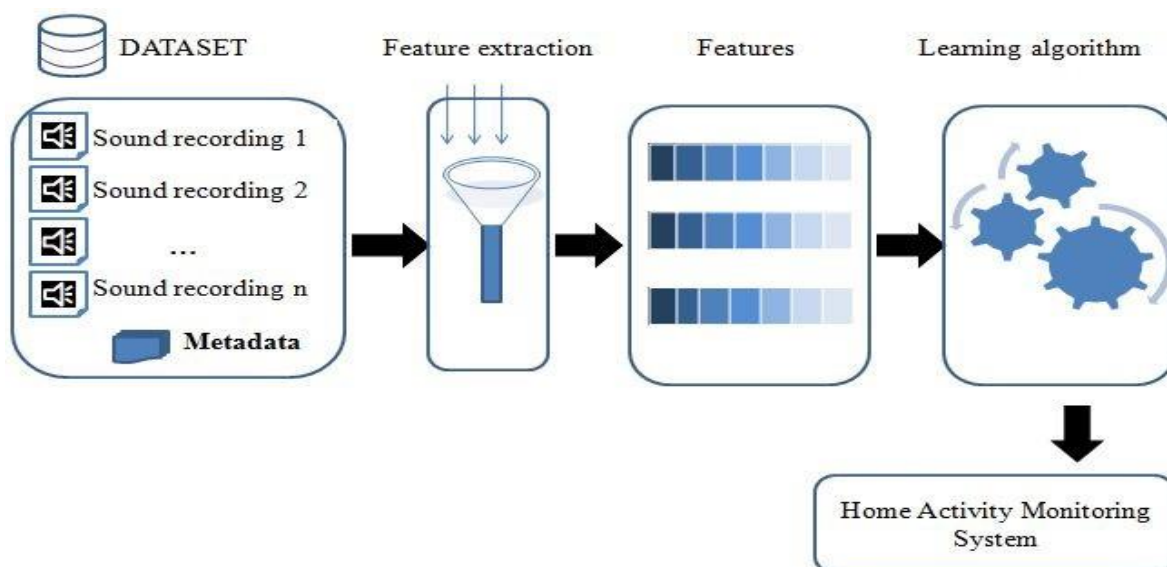


Figure 1.1: HAM System mechanism.

To achieve a smart home audio-based monitoring system, a number of challenges exist in many aspects. Such challenges include data acquisition about the different activities practiced at home. Like any other SED system, **Data collection** is a crucial step in the process of HAM. The collected audio data should be recorded in conditions which mimic the human attitude i.e. data should be as close as possible to reality by providing samples of all home activity classes necessary to enable the machine learning models to learn parameters [8]. Data includes **audio records** and the corresponding **metadata** required for the **supervised learning** approaches due to the class labels it contains [8]. **Metadata** usually references manually **annotated data** collected during the data collection process, it provides information about each activity class within the audio sound. In this Chapter, we present the theoretical side of sound representation and signal processing principles. It is organized as follows: Section 1.2 provides a short definition of sound recording and some useful notations for the rest of the Chapter. Section 1.3 depicts what a multi-channel audio is. Section 1.4 explains the transformation process of the audio signal from time to frequency domain. Section 1.5 deals with the feature engineering stage for transforming the signal into a suitable representation. The last Section 1.6 is dedicated to describe the **feature extraction** techniques that are most commonly used. The last Section is dedicated to show the effect of normalization on features.

## 1.2. Sound recording

Signal or **Sound** is defined as vibrations that travel as waves through the air or another medium and can be heard when they reach a person's or animal's ear [9]. It is also defined as a physical quantity that varies in function of time. The manipulation of this information involves the acquisition, storage, transmission, and transformation. Therefore, sound has two main properties of vibrations: the **amplitude** and the **frequency**. Amplitude is important when balancing and controlling the loudness of sounds, such as with the volume control on a CD player, while Frequency is the speed of the vibration. A given **sound** signal can be divided into a set of possible frequencies (  $f_k$  ) and  $N$  samples with:

-  $f_k \in \mathcal{F}$  denotes the  $k$ th frequency, Where,  $k = 0 \dots N - 1$ .

-  $n_i \in \mathcal{N}$  denotes the  $i$ th sample of the signal.

- $D_k$  denotes the amount of the  $k$  frequency in the signal.
- $D_m$  denotes the amplitude of the signal at the  $m$  sample.
- $M$  is the duration in samples of the audio **frame** (the total number of samples in each frame).
- $m$  the  $i$ th sample of the **frame**.

**Audio signal** quality requires frequency bandwidth to be more than 16 kHz and dynamic range to be more than 80 dB. This will decide the minimum sampling frequency and its minimum coding bit number [9]. In addition, physical limitations of the recorder restrict the selection of the sampling frequency.

### 1.3. Multi-channel audio recording

**Multi-channel recording** means recording more than two separate channels of audio at once on the same computer, synchronized to each other. The provided samples are multi-channel audio segments acquired by multiple microphone arrays at different positions. This means that spatial properties can be exploited to serve as input features to the classification problem. However, using absolute localization of sound sources as input for the detection model is doomed to not generalize well to cases where the position of the microphone array is altered [10] whereas, the use of multi-channel audio signals can be a way to combat overfitting. Typically, this always requires the use of appropriate hardware and drivers with Windows exploitation systems, with recording software that can work with that hardware/drivers combination. For a recorder that is more suited to recording many microphones in the same room in a combined synchronous form, we consider using the MSRS Conference Recorder.

### 1.4. Time and frequency representation

A Time–Frequency representation (TFR) is a view of a signal represented over both time and frequency [11]. Recent researches have proved that human ear resolution varies along the frequency axis containing also non-linear-frequencies. That is, non-linear-frequency scales have been introduced in an attempt to mimic human perception. Thus, TFR such as



Short Time Fourier Transform (STFT) were designed mainly according to mathematical rules leading to linear-frequency scales. The Figure below represents the frequency in function of time and amplitude in function of frequency.

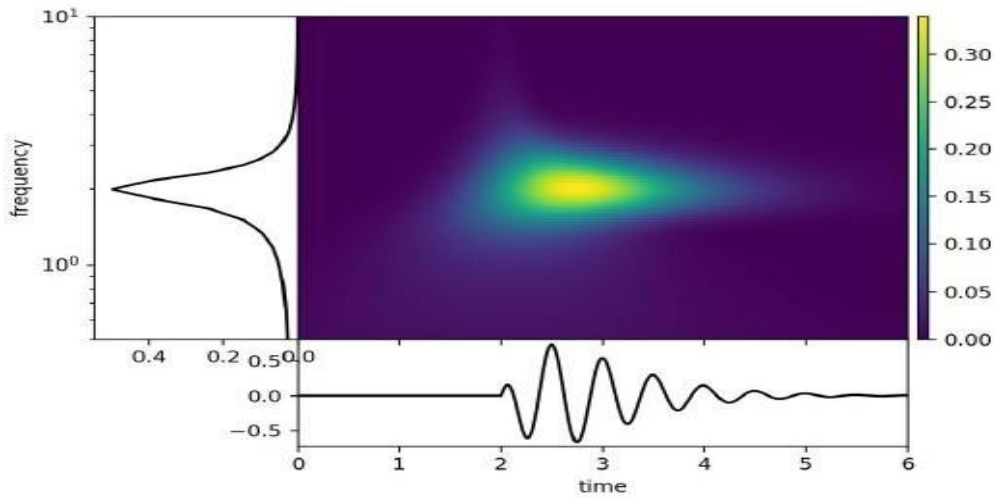


Figure 1.2: Frequency in function of time [11].

### 1.4.1. Fourier transform

Virtually, every entity in the world can be described via a waveform (a function of time); for instance, sound waves. In order to find the different frequencies that are present in a signal. We apply the Fourier Transform to turn a function of time into a function of frequency by summing up its sinusoidal or complex exponential components. The representation of magnitude as a function of frequency is known as the **spectrum of a signal** (See more details in section 1.6.3) [12]. In order to make it possible for an audio signal to be stored and processed, we deal with discrete time signals. These signals are obtained by sampling the original audio sequence at uniformly spaced times with a specific sampling rate. The famous form of the Fourier Transform used to determine the spectrum from the discrete time signals is known as the discrete Fourier Transform (DFT) and it is given by the following formula:

(1.1)

$$D_k = \sum_{ni=0}^{N-1} D_n e^{\frac{-j2\pi Kni}{N}}$$

However, the DFT algorithm has a complexity of  $O(N^2)$ , whereas, the Fast Fourier Transform (FFT) implementation has a quasi-logarithmic complexity  $O(N \log_2 N)$ . For this reason, the FFT implementation is commonly used in practice [13].

## 1.5. Feature engineering

Feature engineering is the process of extracting features from raw data. These features can be used to improve the performance of machine learning algorithms [14]. Feature engineering mainly has two goals:

- Preparing the proper input dataset, compatible with the machine learning algorithm requirements.
- Improving the performance of machine learning models.

**Feature extraction** is one of the most significant factors in audio signal processing [15,16] it is used to provide a numerical representation of the signal that is relevant for machine learning. All classification systems employ a set of features extracted from the input audio signal. Acoustic features can be classified into 3 categories: temporal features, spectral features and prosodic features. Temporal features are represented as amplitude flux within time, they are extracted directly from the audio signals [16]. However, audio signals rely on spectral / cepstral features (see more details in Section 1.6). Spectral features are obtained by converting the time-based signal into the frequency domain using Fourier Transform. Moreover, we can obtain other features as: spectral centroid, spectral flux, spectral envelopes, spectral roll-off, etc. [17].

Whereas, some techniques (Framing and windowing) have to be applied on the audio signal before performing **feature extraction**.

### 1.5.1. Framing

Framing allows the decomposition of a sound signal into a series of overlapping **frames** (in order to capture the signal in a quasi-stationary state) [17]. The rationale behind this step is that frequencies in a signal change with a rapid rhythm over time making the signal non-stationary. That is, we can safely assume that frequencies in a signal are **stationary** over a very short period of time in a short time interval (generally **frames** from 10

to 20 ms). In other words, signal is **stationary** if it has the same frequency over a period of time, the repeating cycles of a frequency don't undergo into variation and remain **stationary**. Figure 1.3 shows Signal Framing.

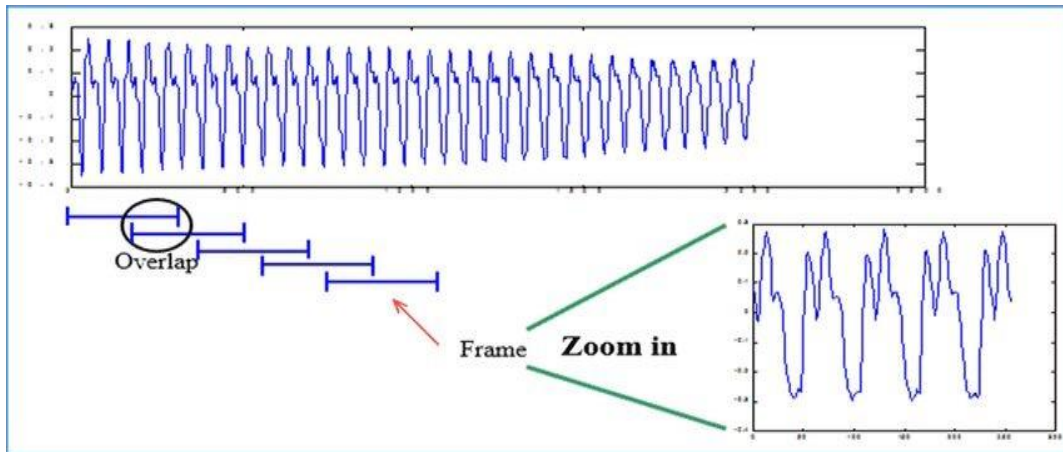


Figure 1.3: Signal framing.

### 1.5.2. Windowing

The frame blocking procedure consists essentially dividing the signal into short **frames** of N samples [18]. We apply a window function to each frame in order to reduce the effect of spectral leakage at the **frame** boundaries when sudden changes in frequency occurred. The most trivial function is the **Hamming window** function. Each sample of the **frame** is multiplied with a **Hamming window** taking the following values:

$$\begin{aligned}
 & 0.54 - 0.46 \cos\left(2\pi \frac{n}{N} \right) \quad / \quad , \\
 & 0 \leq n \leq N-1 \\
 & 1 \\
 & h(n) = \frac{1}{2} \left( 1 + \cos\left(2\pi \frac{n}{N} \right) \right)
 \end{aligned}
 \tag{1.2}$$

The shape of **hamming window** function is represented by Figure 1.4.

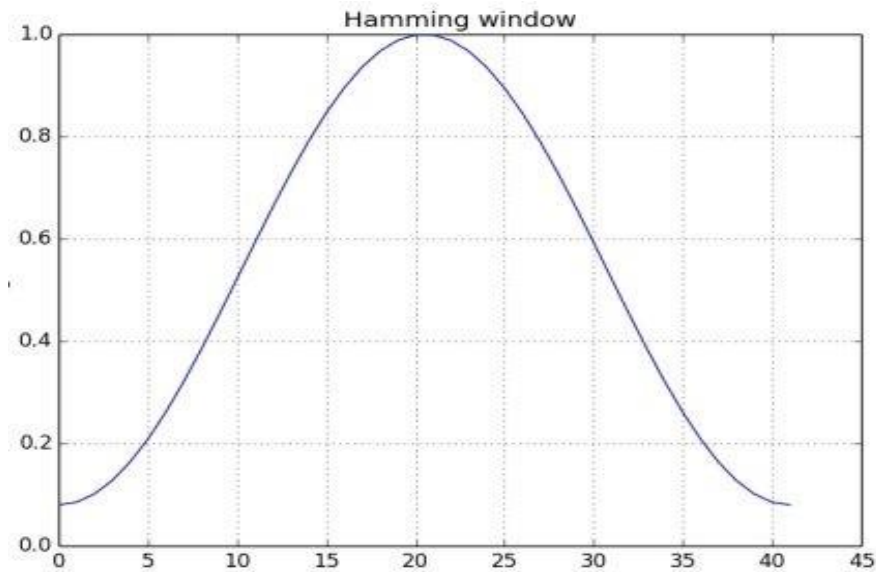


Figure 1.4: Hamming window.

## 1.6. Spectral features

Spectral/ Cepstral features allow us to extract the spectral energy distribution in the audio signal. Their representation is computed based on two main processes The Fourier transform for converting TFR domain and logarithm, which will grant the identification of the basis frequency elements of an audio signal [19]. The most frequently used **feature extraction** techniques are **Log Mel-band energy** and **Mel-Frequency Cepstral Coefficients** (MFCCs), this is due to the fact that they mimic the human auditory perception that focuses on magnitudes of frequency components. The steps involved for their extraction are similar except an extra step (Computing DCT) while computing the MFCCs.

### 1.6.1. Log Mel-band energy features

**Log mel-band energies** are acoustic features used for signal processing. The process of the **Log Mel-band energy** extraction consists of a signal that gets sliced into frames and a window function is applied to each frame; afterwards, we perform a Fourier transform on each frame (more specifically a STFT) and calculate the filter banks, eventually the **Mel-band energies**. After computing the logarithm of the **Mel band energies**, a final step is mean normalization (discussed in detail in section 1.7); each energy band is normalized by subtracting its mean and dividing it by its standard deviation [4]. A general architecture of the Log Mel-band energies extraction process is represented by Figure 1.5.

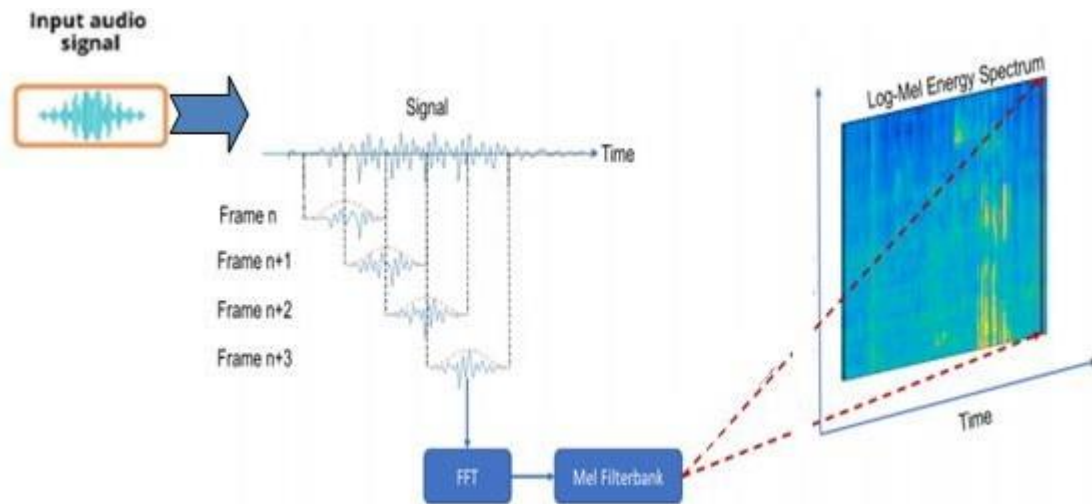


Figure 1.5: Log Mel-band Energy extraction process [20].

### Mel filter banks

A filter bank is an array used to separate the audio signal and eventually produce a bunch of frequency bands. It is a simulation of the mel scale in human ear resolution. The mel scale indicates how to space the filter banks and how the frequency bands are used to quantify the ability of the human ear to distinguish between frequency tones by being more discriminative at lower frequencies and less discriminative at higher frequencies as follows: (frequencies that are in 0-1000 Hz range linearly and above 1000 Hz the perception becomes logarithmic) [3,21]. Energies within each frequency band are summed and their logarithm is taken as an estimation of how much energy exists in that band. We can convert between Hertz and Mel using the following equations:

$$mel = 2595 \log_{10} \left( 1 + \frac{frequency}{700} \right) \quad (1.3)$$

$$frequency = 700 \left( 10^{\frac{mel}{2595}} - 1 \right)$$

Each filter in the filter bank is triangular having a response of 1 at the center frequency and decreases linearly towards 0 until it reaches the center frequencies of the two adjacent filters where the response is 0, as shown in Figure 1.6

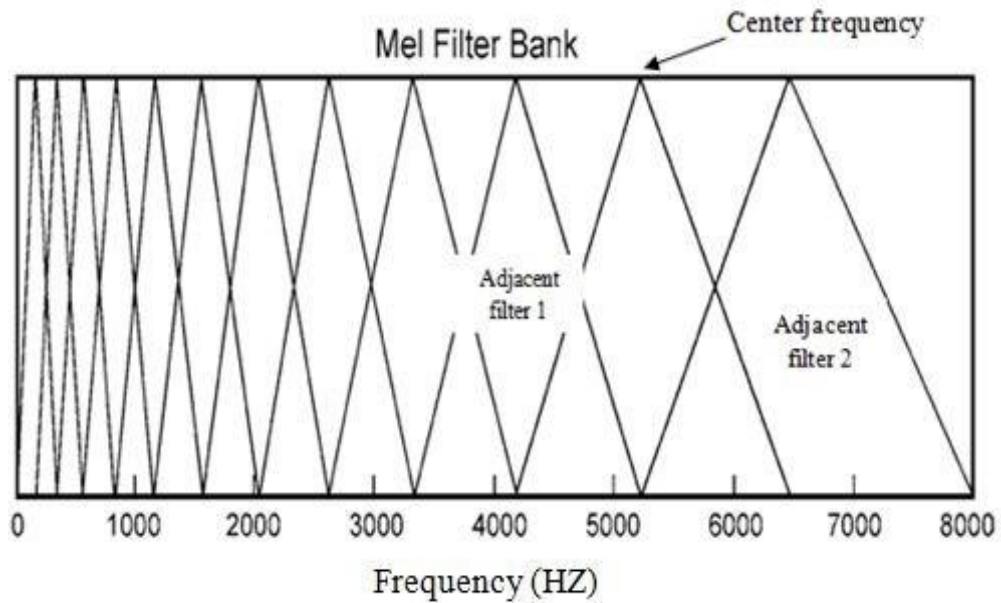


Figure 1.6: Filter bank on a male scale [22].

### 1.6.2. Mel Frequency Cepstral Coefficients (MFCC)

The Mel-Frequency Cepstral Coefficients (MFCCs) were introduced as a type of cepstral representation of audio signals. MFCCs are the result of Discrete Cosine Transform (DCT) applied on **Log Mel-band Energy** features in order to decorrelate the filter bank coefficients.

### 1.6.3. Spectrogram features

Spectrogram is the visual representation of the signal; it displays the amplitude of the frequency components as it changes with time. The spectrogram can be created using several mathematical algorithms, which include FFT. In the figure below, we introduce a spectrogram features sample using FFT.

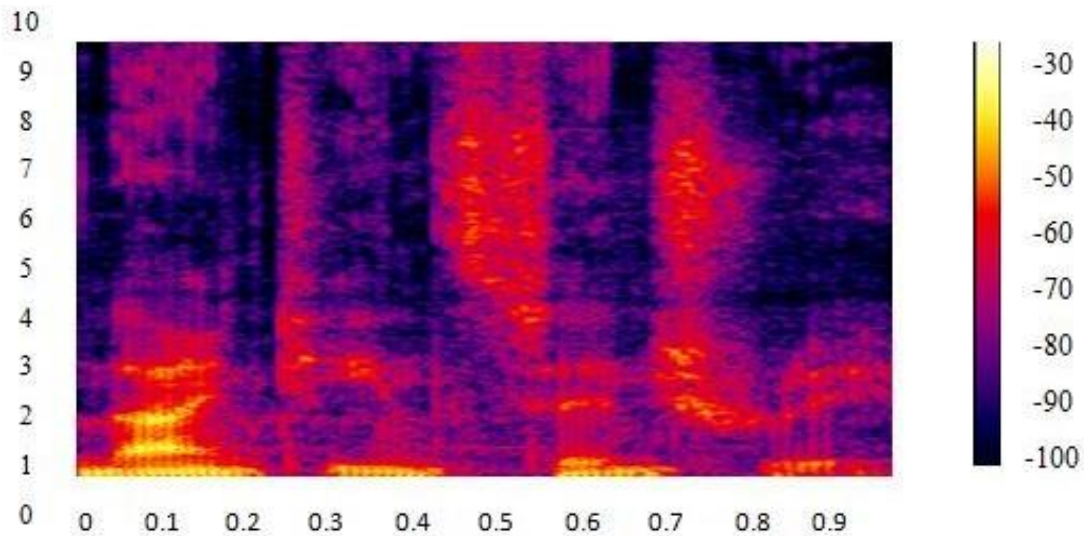
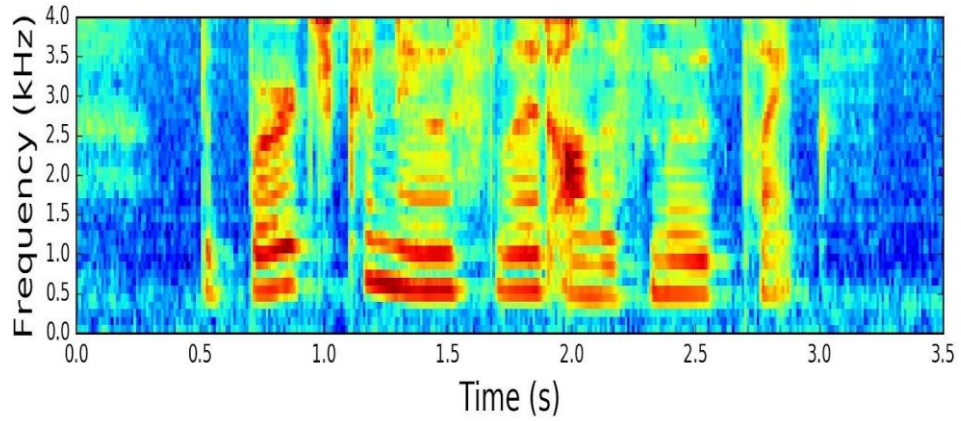


Figure 1.7: Spectrogram feature sample [22].

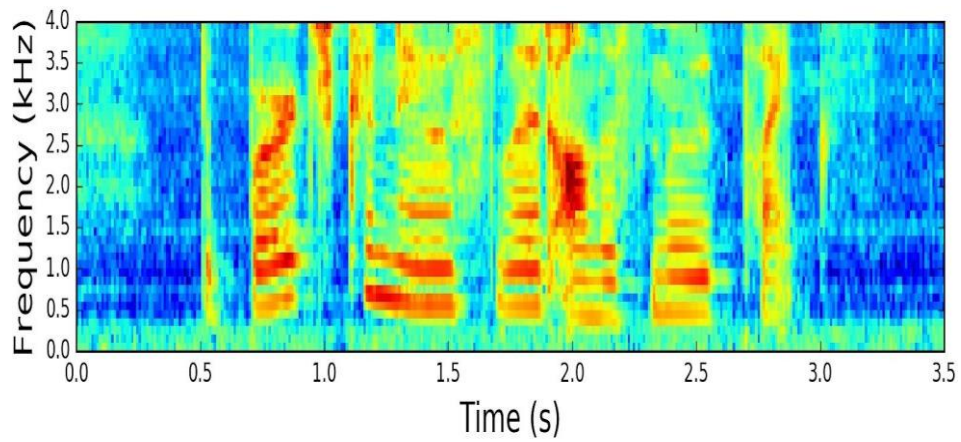
The Spectrogram shows how the energy in different frequency bands changes over time. Frequencies are shown increasing up the vertical axis, and time on the horizontal axis. The legend to the right shows that the color intensity increases with the density; **louder events** (activities) are indicated by brighter colors (yellow, orange, dark orange, purple) **and quieter** events (breaks in activities) are indicated by darker colors (blue/dark blue/black).

## 1.7. Mean Normalization

Mean normalization is used to balance the spectrum and improve the Signal-to-Noise ratio (SNR) by rescaling data to have values between 0 and 1; we can simply subtract the mean of each coefficient from all **frames** and divide it by its standard deviation [22]. The Figure below shows the effect of mean normalization on spectrogram features.



**Spectrogram of the signal**



**Normalized features**

Figure 1.8: Mean Normalization [22].

## 1.8. Conclusion

In this Chapter, we have reviewed the basics of sound representation and signal processing that are necessary for the comprehension of this thesis. We also presented several types of features that exist in literature. These features are required to be an input for the learning stage. The next Chapter will deal with exposing machine learning basic concepts.



# Chapter 2: Machine learning for Home Activity Monitoring

## 2.1. Introduction

**Machine learning** is a field of study that gives computers the ability to learn without being explicitly programmed [23]. However, **deep learning** is a subset of **machine learning** which consists of a set of models known as neural networks. These models need a large amount of data to perform well. **Machine learning** is also defined as programming computers to optimize a performance using example data or past experience instead of directly writing a computer program to solve a given problem. There are three major categories of **machine learning**: supervised, unsupervised and semi supervised learning. We focus on the **supervised learning** approach as it is most frequently used for the analysis of home activities. It consists of building models to learn a mapping between the extracted features and sound class labels predefined in the reference **metadata**.

This Chapter is structured as follows: Section 2.1 is a short introduction to **machine learning**. In Section 2.2 we present fundamentals of classification approaches used for the HAM purpose. In Section 2.3 we put the highlight on supervised learning by providing a definition for the Neural Network classifier used in this study. After that, we describe some evaluation metrics that are well known for evaluating learning models and cover some challenges facing the HAM researches in Sections 2.4, 2.5 respectively; then, we discuss in Section 2.6 several state-of-the-art techniques that are related to our study.

## 2.2. Classification

The most common task in **machine learning** is **classification** [24]. It is concerned with the problem of attributing class labels to unseen objects. In case of HAM, an object is known as a feature vector. The resulting model known as classifier or learner enables us to predict the class label of the unseen object.

We consider the problem of classification of home activities as a mapping from the patterns features space  $\mathcal{X}$  to the class labels space  $\mathcal{Y}$  where an instance (also called pattern,

sample and object) is characterized by a feature vector  $x = [x_1, x_2, \dots, x_n]$  and by its class label  $y \in \{1, 2, \dots, K\}$  [25], with each feature is attributed to one and only one class since we are monitoring monophonic activities. In supervised learning, the role of any classification algorithm is to learn predictive model from a set of data samples which have been already labeled, using a mapping function  $\{y = f(x)\}$  which takes input a feature vector  $x$ , some parameters and produces an output  $y$ . This output can be either:

- A simple class label.
- An Oracle output defined as a Boolean vector  $V = [v_1, v_2, \dots, v_n]$  with  $n$  is the size of the training dataset,  $v_i = 1$  if the classifier correctly classifies instance  $x_i$  and 0 otherwise.
- A probability vector  $K = [k_1, k_2, \dots, k_m]$  with  $K_i \in [0, 1]$  over all class labels.

**Classification** task is divided into two main phases: The classifier learns the mapping input to output in a **phase of training**, and then makes predictions for new inputs in a **testing phase** [17].

In the training phase, the model is trained on the training dataset [26, 42] using a supervised learning method (e.g. gradient descent) (more details can be found in Section 2.3.1). In practice, the training dataset often consists of pairs of an input vector and the corresponding output vector (target or label).

The test dataset is used to provide an unbiased evaluation of a final model fit on the training dataset [27]. In this phase, the produced model is used to predict the class labels of unseen objects. If the data in the test dataset has never been used in training, the test dataset is called a holdout dataset.

The classification step has two major issues: overfitting and underfitting. A model overfits if it fits the training data **too well** and there is a poor generalization of new data. On the other hand, a model underfits if it fits the training data is not enough to get important results. The main goal in this phase is building a model with favorable fitting and gives good performances. A further validation dataset can be used for regularization by early stopping: stop training when the error on the validation dataset increases, as this is a sign of overfitting to the training dataset [28,42].

There are multiple classifiers in the machine learning community such as Neural Networks, K-nearest neighbors, Random forest, Adaboost and Support vector machines. The aim of any one of them is to find the model parameters that produce the best results. We measure the performance of a classifier by using multiple evaluation metrics. In the next Section, we put the highlight on the Neural Network classification approach that is a supervised approach providing probability outputs.

### 2.3. Neural Networks

Neural networks (NNs) are computing systems that process information by their dynamic state response to external inputs [29]. NN is known as a network of interconnected nodes called **neurons**. Nodes are joined to each other with links having each one of them a weight  $w_i$  which determines the strength and sign of the connection. A link from one unit to another is directed and serves to propagate the activation [30]. Each node computes the sum of input weights, and then it applies an **activation function** (Sigmoid, ReLU, Softmax) on this sum to derive the output [31]. Usually, Traditional Neural networks are composed of one input layer, one hidden layer, and one output layer, more than one hidden layer qualifies a network as a deep learning network. The Figure below shows a simple structure of a Multi-Layer NN comprising one input layer, hidden layers and finally the output layer.

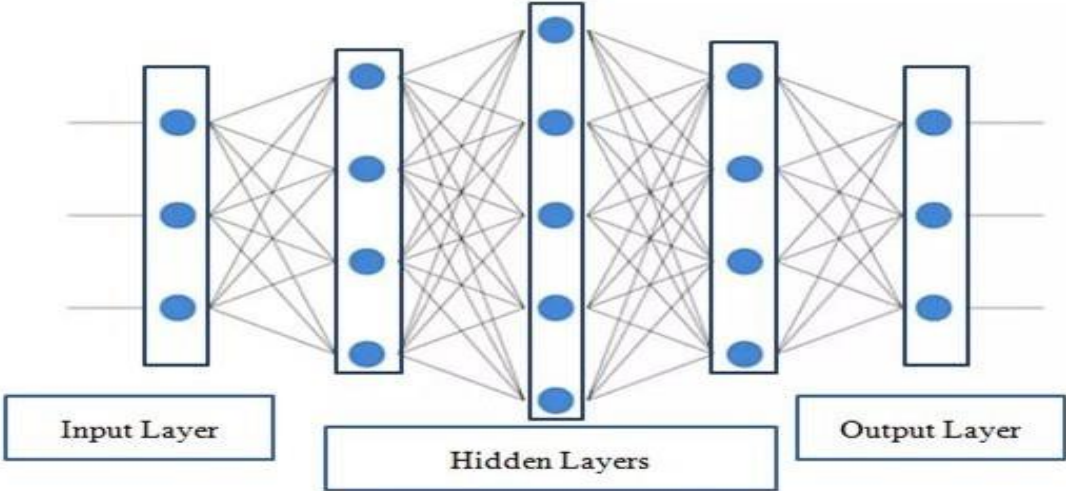


Figure 2.1: Multi-Layer Neural Network.

Multiple models of Neural Networks exist: Recurrent Neural Network, Convolutional Neural Network, Deep Neural Network and Deep belief network. For the purpose of modeling a HAM system, this thesis focuses on Convolutional Neural Networks.

### 2.3.1. Convolutional Neural Network

Convolutional neural networks (CNNs) are biologically-inspired from the animal's visual cortex that is composed of a collection of neurons connected to each other and organized in hierarchical layers [30]. CNNs are so far used for analyzing images; their utilization is expanded for spectral features which are considered as input features for the model. Any CNN is composed of a Convolutional layer, Activation layer, Pooling layer, and Dense layer (also known as fully connected layer), the architecture of the CNN is illustrated in Figure 2.2.

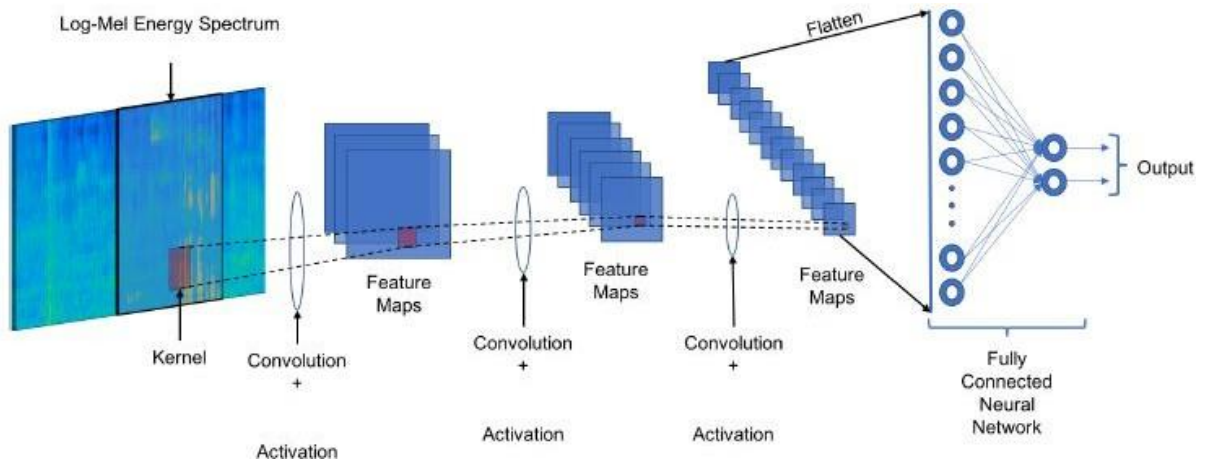


Figure 2.2: Architecture of a CNN [30].

For the rest of this Chapter, we denote:

- $x_i$  as a single features value within the feature matrix  $X = \{x_1, x_2 \dots x_n\}$ .

-We also define  $y_i$  as a single probability value that matches the input within the normalized feature matrix  $Y = \{y_1, y_2 \dots y_m\}$ .

The **Log-Mel Energy spectrograms** are considered as input feature vectors for the learning stage. The latter go through the different layers of CNN starting with the

**convolutional** layer followed by the **activation** and **pooling** layers. The whole set is grouped in a **Hidden Layer**; the last layer is the **dense** layer (or fully connected layer).

### a. Convolutional Layer

The first layer is the **Convolutional** layer, for each **convolutional** layer, we need to specify the number of filters  $n$  (also called filter matrix and kernel). A filter is a small matrix of 2 dimensions which slides the entire input, this sliding is called **convolving**. The filter moves over each portion of the input  $x_i$  with a certain stride value. It consists of a product between the matrix representing the input feature  $X$  and the filter matrix. This process is repeated until the entire input feature  $X$  is traversed. The resulting matrix is known as convolved feature matrix which contains a set of feature values that matches the input and a set of zeros that represents the irrelevant parts where there were no matches. It is worth noting that normalization of the convolved feature matrix is done using the **Softmax** activation function. The process of **convolution** is depicted by Figure 2.3.

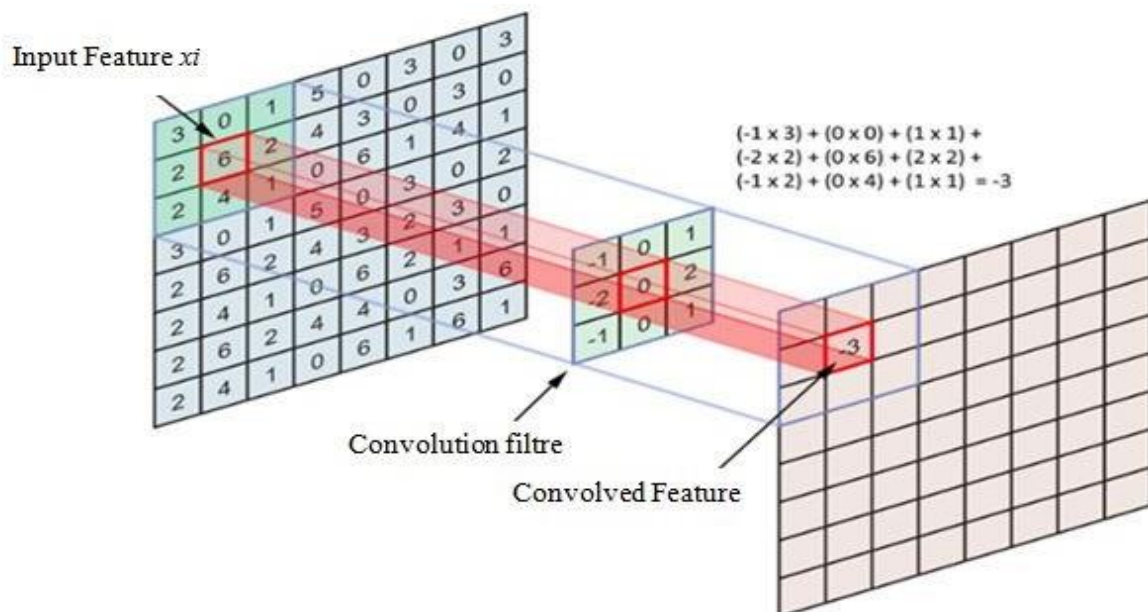


Figure 2.3: Convolution process [32].

### b. Activation Layer

At this level, an **activation function** is applied on the entire convolved feature matrix to enable the model to classify nonlinear data. Three **activation functions** are mostly used with

NNs: sigmoid function, softmax, and ReLU. It is worth mentioning that any type of convolution involves a ReLU operation, without that the network won't achieve its true potential [32].

### Rectified linear Units activation function

Rectified linear Units activation (ReLU) function is the most commonly used function. It is a special implementation that combines non-linearity and rectification layers in CNN. The output of ReLU function is a 0 if the input is less than 0, and raw output otherwise. That is, if the input is greater than 0, the output is equal to the input. If we consider a single node in a NN model assuming it has two inputs called  $x_1$  and  $x_2$ , their weights  $w(x_1)$  and  $w(x_2)$  into our node are 2 and 3 respectively. So the node output is  $f(2x_1+3x_2)$ . We'll use the ReLU function for our  $f$ . So, if  $(2x_1+3x_2)$  is positive, the output value of our node is also  $(2x_1+3x_2)$ . If  $(2x_1+3x_2)$  is negative, the output value of our node is 0. It is defined as:

$$LU(x_i) = \max(0, x_i) \tag{2.1}$$

There are many similar alternatives which also work well. The Leaky ReLU is one of the most well known, it is the same as ReLU for positive numbers but instead of being 0 for all negative values, it has a constant slope (less than 1). So the values in the final feature matrix are not actually the sums, but the ReLU function applied to them.

### Softmax Function

Softmax function is applied on each  $x_i$  of the feature matrix. It computes the probabilities of each activity over all possible activities. The output is a normalized feature matrix  $y_i$  representing probability distribution for each predefined class (it measures the probability that any of the classes are true). It is defined by:

$$y_i = \text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{k=0}^n e^{x_k}} \tag{2.2}$$

Softmax function is generally used in the Output Layer to flatten and normalize the convolved and maxpooled matrix.

### c. Pooling Layer

A **pooling** layer is a new layer added after the **convolutional** layer. In the **pooling** layer the input matrix is reduced into smaller one by extracting only the most relevant features. This enables us to reduce the number of parameters, which both shortens the training time and combats overfitting. **Pooling** layers downsample each convolved feature matrix independently, reducing the height and width, keeping the depth intact.

The most common type of pooling is **max pooling** which takes the max value in the **pooling** window. Contrary to the **convolution** operation, **pooling** has no parameters. It slides a window over its input, and simply takes the max value in the window. However, similar to **convolution** we specify the window size and stride [32].

### d. Dense Layer

The last Layer is the **dense** layer or the fully connected layer which aims to classify the inputs by flattening them into a single vector of probability values  $\in [0,1]$  that indicates either the feature belongs to the label or not. It outputs a feature matrix. Remember that the output of both convolution and pooling layers are 3 dimensions volumes, but a fully connected layer expects a 1 dimension vector of numbers; that is why we flatten and normalize the output matrix of the final pooling layer into a vector of probability values whose total sums up to 1 using the **softmax function**. This vector is used to train the model by applying the **gradient descent** process which consists of finding the weights that minimize the loss between the actual and the predicted output in the training set using the cross entropy loss function over a series of epochs during the training phase until convergence. This optimization algorithm is said to converge when the gradient gets closer to 0 and remains stable. After the learning step, the dense layer outputs a probability distribution for each predefined class.

Due to the lack of data to train CNN models, these letters generally use methods like dropout (widely detailed in Section 3.5.2) that aim to improve the performance by increasing the generalization ability of the learning model.

## 2.4. Evaluation of HAM systems

### 2.4.1. Cross validation

Cross validation is one of the techniques used to test the effectiveness of **machine learning** models, it is a procedure used to evaluate a model if we have a limited dataset in order to learn a hypothesis from a training set and to measure its generalization error on a test set. Numerous techniques have been introduced such as k-fold cross validation. This latter is an iterative approach: during iteration  $k$ , it randomly divides the set of observations into  $k$  groups or folds, of approximately equal size. Fold  $k$  is treated as a testing set, and the remaining  $k - 1$  folds assigned as a training set. This procedure is repeated  $k$  times as shown in Figure 2.4, the data is divided using a 4-Fold cross validation.

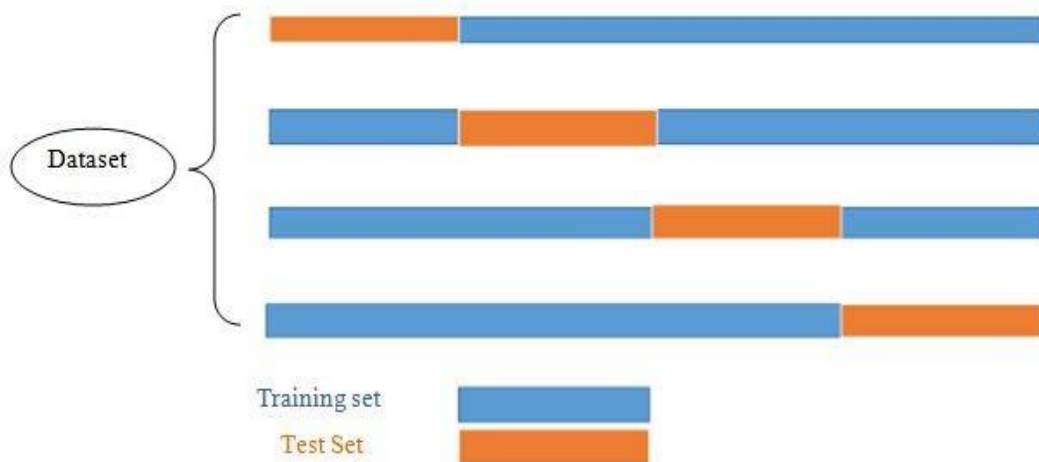


Figure 2.4: 4-Fold Cross Validation [43].

### 2.4.2. Measures of Performance

Evaluation of HAM systems is done by comparing the system output with **the reference annotations** available for the test data. Numerous metrics are introduced to be used including: precision, recall and F-score.



### 2.4.2.1. Confusion matrix

The Confusion matrix is a metric used for detecting the correction of a model. It is a matrix of  $n \times n$ , where  $n$  represents the number of classes. The row dimension is called Ground truth, whereas the column is known as Predicted. Confusion matrix can contain values in (0,1) which combination can be either TP, FP, TN or FN where:

- **True Positives (TP):** Cases where actual class of the data is 1 (True) and the predicted class is also 1 (True).
- **False Positives (FP):** Cases where the actual class of the data is 0 (False) and the predicted is 1 (True).
- **True Negatives (TN):** Cases where the actual class of the data is 0 (False) and the predicted is also 0 (False).
- **False Negatives (FN):** Cases where the actual class of the data is 1 (True) and the predicted is 0 (False).

### 2.4.2.2. Precision and Recall

Precision is the fraction of predicted positives which are actually positive (correctly detected events). Recall is the fraction of actual positives which are correctly predicted (missed detections). We can calculate them from the confusion matrix using the equations 2.3 and 2.4.

$$Precision = \frac{TP}{TP + FP} \quad (2.3)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.4)$$

Precision, Recall and F-score are **segment-based statistics** calculated using **class-based averaging**. In **class-based averaging** (macro-averaging), intermediate statistics are accumulated separately for each event class and are used to calculate class-wise metrics. Overall performance is then calculated as the average of class-wise performance.

### 2.4.2.3. F-Score

F1 Score measures how many instances it classifies correctly (Precision), as well as if it does not miss a significant number of instances. It is the average between precision and recall. F1 Score is given by the following formula:

$$F1=2 \times \frac{Precision + recall}{precision \times recall} \quad (2.5)$$

We conclude that the greater the F1 Score, the better is the performance of our model.

## 2.5. Challenges within HAM Systems

To achieve a robust activity monitoring system in a smart home, a number of challenges still exist in many aspects of the development procedure [3].

### 2.5.1. Overlapping events

Real-life sound recordings typically have many overlapping sound events, making it hard to recognize each event within a sound recording. Monophonic sound event detection in a domestic environment deals with the most prominent event at a time instance while polyphonic detection tackles the situations where multiple sound events happen simultaneously is causing a real challenge to HAM monophonic systems [3].

### 2.5.2. Human attitude

Human attitude cannot be expected all the time; thus, he can perform activities that are new, unknown or rare to the trained model [3]. These activities are often unlabeled within the training dataset, which provokes many underfitting issues during classification.

## 2.6. Related Work

Numerous works have focused on monitoring acoustic activities within homes using multichannel-audio signals; each one of these works uses different methods of **feature extraction, machine learning**.

In their works, W. Zhang et al. [33] have proposed models for HAM based on Gated Neural Networks (GCNN). The GCNN works as follows: the output of the convolutional layer is divided into two parts with the same size denoted  $x1$  and  $x2$  respectively. Then  $x1$  passes through an activation function and multiplies with  $x2$ . Then two dense layers are used to combine extracted features and output nine scores. The results show that the performance is relatively high-ended.

Chew et al. [34] proposed a system for Health monitoring addressed for elderly people. It consists of three different models from different audio channels. The proposed system can classify different domestic activities effectively with a relative f-score.

In the same context, Nakadai et al. [35] have proposed a partially-shared CNN, which is a multi-task system that contains a common input and two output branches: a classification branch which outputs the predicted class and a regression branch, which outputs a single-channel representation of the multi-channel input data. Authors tried to improve classification performance with this system by training classification and regression together.

Furthermore, Alon, A et al. [36] have suggested a multi-channel processing to classify the audio signals to one of the predefined classes. The 4 transforms of each of the two layers are combined together. Each of the fused layers is processed in parallel by two neural networks (NN) architectures, RESNET and long short-term memory (LSTM) network.

Table 2.1: HAM related work.

Ref	Feature extraction technique	Classifier	Performance by F1-Score
[33]	log-mel energies	GCNN	<b>89.73%</b>
[34]	MFCC	CNN, LSTM,ensemble	<b>92.19%</b>
[35]	log-mel energies	Partially Shared CNN	<b>89.94%</b>
[36]	Scattering Transform	LSTM, CNN, ResNet	<b>1<sup>st</sup> Layer 85.67 %</b> <b>2<sup>nd</sup> Layer 85.82 %</b>

## **2.7. Conclusion**

Across this Chapter, we have reviewed the crucial concepts of classification. The main goal of our work is to conduct Machine Learning experiments for acoustic event detection within smart homes. First, we have presented what classification is, and then we highlighted the CNN classification models as the main model required for our work. We have also introduced famous measures to evaluate the efficiency of learning models. Then we have talked about related work to HAM in the last Section. In the next Chapter, we will introduce the experimental setup and describe the general environment deployed for our experiments. Then, we expose and discuss the obtained results.

## Chapter 3: Experimental setup and results

### 3.1. Introduction

In this Chapter, we present the full setup used to grant our experiments and the experimental results. Firstly, we present our dataset in Section 3.2. Secondly, we present the tools conducting to realize the experiments in Section 3.3. Thirdly, we describe the implemented HAM system using the **Log Mel Band Energies** for feature extraction and CNN for **classification** in Section 3.4. Fourthly, in Section 3.5 we describe the experimental setup of the system. Section 3.6 of this Chapter is dedicated to present and discuss the experimental results. Finally, in Section 3.7 we give the consumed time of experimental studies during training and testing; then we conclude the Chapter.

### 3.2. Dataset Presentation

#### a. Dataset Description

In this study, we are using the DCASE2018 dataset [10,37]. It is a huge dataset (**87.5 Gb**) composed of sound recordings of one person over a period of one week in his home. The data composing the dataset was collected using a network of 13 microphone arrays settled over the entire home. The recordings were split into audio segments and segments containing more than one class were left out. This means that each segment represents one activity and no overlapping was detected (**Monophonic segments**). Each audio segment is a multi-channel recording which contains 4 channels.

The sensor node configuration used in this setup is a linear microphone array. The sampling for each audio channel is done sequentially at a rate of 16 kHz. The data provided to observe performed activities for each sensor node contain recordings from multiple microphone arrays at the same time instant. This means that the performed activities are observed from multiple microphone arrays at the same time instant. The annotation was performed during the data collection; a Smart phone application was used to let the monitored person annotate the activities while being recorded. The person could only select a fixed set of activities. The start and stop timestamps of each activity were refined by using an annotation

software. The pre-processing stage is performed on the set already. The provided dataset contains a validation dataset to evaluate the resulting model. The daily home activities considered for this work are exposed in Table 3.1 along with the frequency of occurrence of the 10s multi-channel segments in the development set [37].

Table 3.1: Frequency of occurrence of 10s segments daily activities in DCASE2018 dataset.

Activity	10 seconds Segments
Absence	18860
Cooking	5124
Eating	2308
Dishwashing	1424
Social activity (Phone call, visit...)	4944
Working (Mouse click, typing...)	18644
Vacuum Cleaning	972
Watching TV	18648
Other (Noise, No relevant activity)	2060
<b>Total number of 10s activities</b>	<b>72984</b>

## b. Dataset Slicing

For the purpose of evaluating our HAM system, our dataset is divided into 4 output folds containing each one, training and testing sets using **4-cross validation**. Data was **shuffled** to generate different combinations in order to ensure that the activities are present in the testing set. Over the 4 iterations of the process, we consider one fold as a testing set and the rest of folders as a training set; then we fit a model on the training set and evaluate it on the testing fold. Once all models are trained, an average F-score is calculated over the results of the 4 iterations.

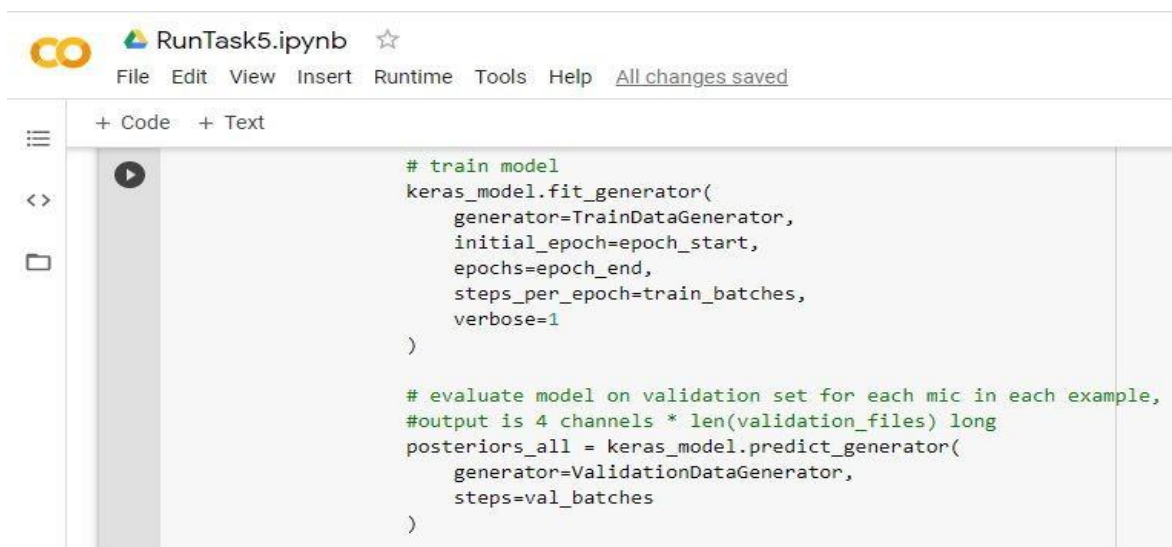
## 3.3. Tools

We have underwritten our experiments using Python (Version 3.6.10) programming language. Python is an object-oriented language which is commonly used for machine learning purposes due to the rich libraries it contains such as Pickle, Librosa [38] and Keras.

- **Pickle**: we have used Pickle library which is used for serializing/ deserializing the structure of audio signal into numeric representation in order to save it for further learning processes.
- **Keras**: The learning part of the code was built on Keras 2.1.5 which exposes a wide variety of machine learning algorithms including CNN. This library is used with **Tensorflow** 1.4.0 as backend and **Scikit-Learn** 0.19.1 (for calculating F-Score.) to conduct the machine learning process [39].

The system is based on the **Dcase\_util** library (0.2.4) and has all needed functionality for dataset handling/storing /accessing features and models, and evaluating the results. Other components like Numpy 1.9.2, Librosa 0.7.2 (for signal processing used for displaying spectral features [38]) are available.

As a first step, we have explored a python-based environment named Spyder 4.1.3 to perform feature extraction then the model is supposed to be trained using an online platform providing a free Cloud service called **Google Colab** [40]. This choice was made because of the huge mass of audio data in the DCASE2018 set assumed with **87.5 GB** in order to cope with the increasing cost in terms of **storage** and **time consumption** while training the model. **Google Colab** notebooks execute python code on Google's cloud server and enable us to visualize the results on Google Drive. Figure 3.1 presents a screenshot of **Google Colab** notebook.



```

# train model
keras_model.fit_generator(
    generator=TrainDataGenerator,
    initial_epoch=epoch_start,
    epochs=epoch_end,
    steps_per_epoch=train_batches,
    verbose=1
)

# evaluate model on validation set for each mic in each example,
#output is 4 channels * len(validation_files) long
posteriors_all = keras_model.predict_generator(
    generator=ValidationDataGenerator,
    steps=val_batches
)

```

Figure 3.1: Screenshot of Google Colab notebook.

### 3.3. System Description

Our system aims to classify multi-channel audio segments (i.e. segmented data is given) acquired by microphone array, into one of the provided predefined classes representing home activities such as “Eating”, “Watching TV”, “Cooking” and “Dishwashing” as illustrated in Table 3.1. In this study, a person living alone at home is considered, and the considered dataset does not contain overlapping activities. Furthermore, the focus in this work is on systems which can exploit features independent of sensor location using multi-channel audio.

Our objective is to experimente the ability of a HAM system to indicate whether the activity is present or not within the given set of multi-channel audio recordings. We have carried our experiments combining the Log Mel-band Energy features with CNN classifier. Figure 3.2 presents the different steps to develop the system.

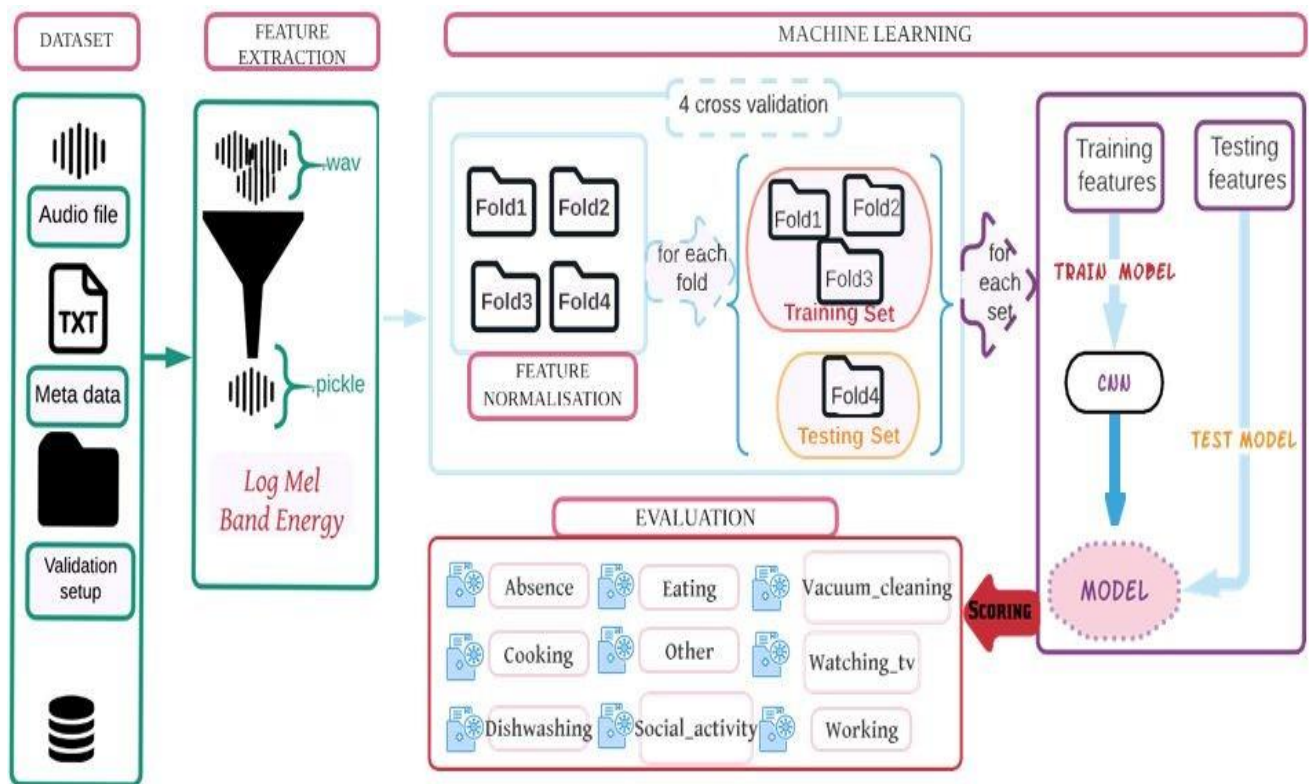


Figure 3.2: Principal steps for conducting HAM system.



### 3.4. Experimental Setup

In this study, we are developing a classifier model that takes a single channel as input. Each parallel recording of a single activity is considered as a different example for this channel during training. The learner is based on CNN. As input, log mel-band energies are provided to the network for each microphone channel separately. In the prediction stage, a single F1-score output is computed by averaging the 4 model outcomes (posteriors) that were computed by evaluating the trained classifier model on all 4 microphones.

#### 3.5.1. Log Mel-band Energy Setup

The parameters used for extracting the Log Mel-band energy features are summarized in Table 3.2.

Table 3.2: Log Mel-band energy features Setup.

<b>Parameter</b>	<b>Value</b>
Frame size	40 ms
Sample rate	16 kHz
Feature Vector length	40 in a sequence of 501 successive frames of 10s
Number of Mel filters	32
Number of Log Mel-band energies	40
Number of FFT	1024
Window function	Hamming_asymmetric
<b>Total number of features</b>	<b>72984</b>

However, we have used the serialized format using the pickle features (. pickle extension) to store our data, as it is smart in dealing with recursive structures like spectrograms. The choice of using pickle library was led by the fact that it can be advantageous for converting a

python object into data that can be transferred over the network, written to a file, or even stored away in a database. When the object is later needed, the Pickle module can convert the serialized data into a regular python object [45]. Figure 3.3 depicts the Log Mel-band energy spectrograms for each sound activity class from our dataset.

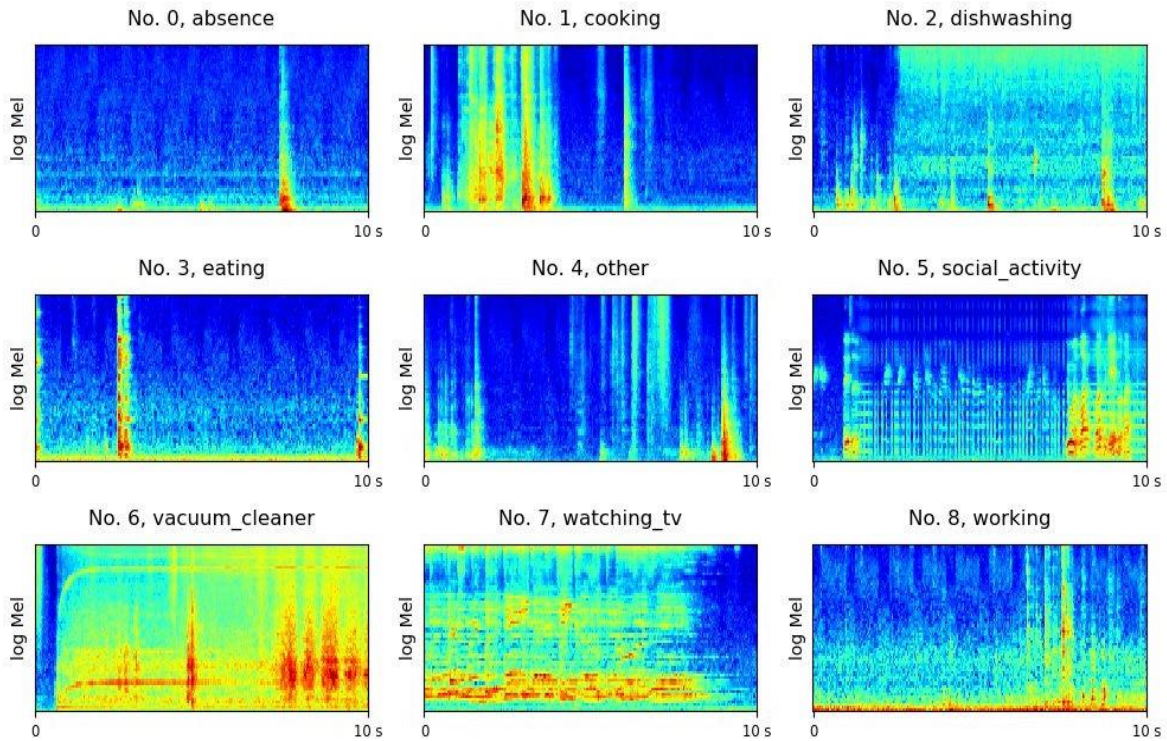


Figure 3.3: Log Mel-band Energy spectrogram features.

### 3.5.2. CNN Setup

We have trained our models on Log Mel-band features using CNN, while varying some hyper parameters such as number of epochs, batch size and Dropout rate. In order to improve the classifier’s performance, we have tested varying the number of epochs. We have set it to either **100** or **500**, while maintaining exactly the same values of the other parameters; in such case we will have two distinct CNNs, we eventually call them CNN100 and CNN500. Table 3.3 provides a description of the CNN setup used for learning stage; some of its parameters are detailed in what follows:

- **Dropout:** In order to address overfitting, we use regularization techniques to improve the generalization ability of our model. The most popular regularization technique in neural networks is called **dropout**. It consists of a set of randomly selected neurons to

be ignored in each epoch of training. This means that their contribution to the activation of neurons is temporarily removed.

- **Batch size:** In order to train our model we need to specify the batch size. It is the number of samples in the input data that will be passed through the network at one time. However, the quality of our model may degrade as we continue on increasing the batch size.

Table 3.3: CNN Setup.

Architecture of CNN	Parameter	Value
<b>Convolutional Layer</b>	Filters	32
	Kernel size	5
	Stride	1
	Axis	Time
	Activation function	ReLU
<b>Pooling Layer</b>	Pooling function	Max Pooling
	Pool size	5
	Stride	5
	Pooling Dropout	0.2
<b>Convolutional Dropout Layer</b>	Filters	64
	Kernel size	3
	Stride	1
	Axis	Time
	Activation function	ReLU
	Convolution Dropout	0.2
<b>Dense Dropout Layer</b>	Number of neurons	64
	Activation function	ReLU
	Dense Dropout	0.2
<b>Output Layer</b>	Input feature matrix	40*501
	Epochs	100/500
	Processing interval	10
	Batch size for each channel	256*4
	Activation function	Softmax
	Classes	9

CNN considers each channel as a single input; thus, the final step consists of Fusion: Output probabilities from the four microphones in a particular node under test are averaged to obtain the final posterior probability using **mean fusion method**. The performance of the model is evaluated over each fold every 10 epochs on a validation subset using the F1-score; the model with the highest Macro-averaged F1-score is picked.

### 3.5. Experimental results

After setting-up our model for training and testing, we present and discuss the experimental results of the case study. Table 3.4 represents the average F1-Score results obtained for this scenario. Each activity is evaluated over 4 folds using class-based averaging. Intermediate F-Scores are accumulated over the 4 folds for each activity class. These latter are used to calculate class-wise F1-Score. An Overall performance is then calculated as the average of class-wise performance.

Table 3.4: F1-Scores (%).

Activity	<i>Log mel-band energies</i>									
	<i>CNN100</i>					<i>CNN500</i>				
	Fold1	Fold2	Fold3	Fold4	<i>Class-wise F1-Score</i>	Fold1	Fold2	Fold3	Fold4	<i>Class-wise F1-Score</i>
Absence	79.86	79.47	84.31	85.01	<b>82.16</b>	85.67	86.81	88.37	90.84	<b>87.92</b>
Cooking	91.15	90.99	90.37	90.08	<b>90.57</b>	95.77	94.91	92.06	96.40	<b>94.78</b>
Dishwashing	55.69	51.84	61.84	64.31	<b>58.42</b>	79.90	69.16	69.29	81.06	<b>74.85</b>
Eating	70.19	73.41	74.14	79.29	<b>74.26</b>	80.79	81.31	83.09	90.72	<b>83.98</b>
Other	23.10	29.45	33.38	35.63	<b>30.39</b>	38.55	44.59	47.82	52.44	<b>45.85</b>
Social_Activity	92.33	84.49	96.59	96.60	<b>92.51</b>	96.11	87.37	95.99	96.77	<b>94.06</b>
Vaccum_Cleaning	98.31	98.85	99.56	100.00	<b>99.18</b>	97.08	100.00	99.34	100.00	<b>99.11</b>
Watching_TV	99.59	99.83	99.40	99.88	<b>99.68</b>	99.24	99.88	98.95	99.92	<b>99.50</b>
Working	74.62	62.91	68.21	77.69	<b>70.86</b>	83.64	80.04	80.92	89.00	<b>83.40</b>
<i>Overall class-based F1-Score</i>	<b>76.09</b>	<b>74.55</b>	<b>78.64</b>	<b>80.95</b>	<b>77.56</b>	<b>84.08</b>	<b>82.67</b>	<b>83.98</b>	<b>88.57</b>	<b>84.83</b>

### 3.6.1. Discussion

Proceeding from the previous results we conclude that the combination of Log mel-band energy features with CNN learner is significantly productive. For 100 epochs, the system performance was relatively high; the F1-Score was estimated with **77.56%**. Moreover, while increasing the number of epochs to 500 we observe that the results have significantly improved for each one of the following classes (Absence, Cooking, Dishwashing, Eating, Other, Social\_Activity and Working) while we observe a decrease in class-wise F1-Score for (Vaccum\_Cleaning and watching\_TV), that is due to the fact that the model is underfitting on these classes when the number of training epochs is increasing, and there is no sufficient amount of data to avoid this issue. The Overall F1-Score has achieved **84.83%**. Therefore, we conclude that the increasing number of iterations is an important factor for improving **the generalization ability of CNNs**. Thus, the greater the number of epochs, the better is the performance. We present in Figure 3.4 a summary of the results and compare CNN100 and CNN500 by class-wise performance. Moreover we present the overall performance of each experiment.

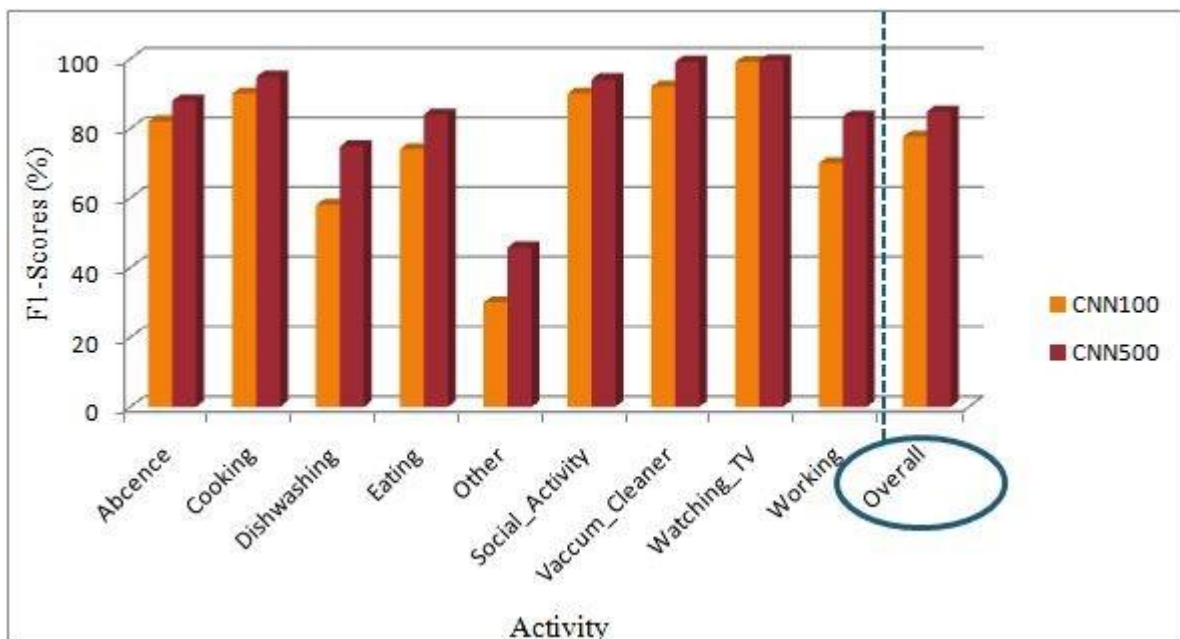


Figure 3.4: F1-Scores (%) For each class and Overall F1-Score for CNN100 and CNN500.

### 3.6.2. Execution Time

The time used to train and test the CNN model is enormous, especially during training as the amount of data is huge. It took around **7 days to train and test the CNN100** model; thus, **48h hours to train a single fold**; whereas, it took **10 days to train and test a single fold from CNN500** which makes the total duration of time **40 days** for CNN500.

### 3.6. Conclusion

We have described through this Chapter the full experimental setup starting from describing the dataset and how it is divided into training and testing datasets. Also, we have depicted a set of tools used to manage our system. Then, we have presented a schema that highlights the important steps carried out to build a HAM system. Afterwards, we have presented the experimental setup. Finally, we have presented and discussed the results of this experimentation. The analysis of the results leads to conclude that Log mel-band energies combined to CNN are significantly productive. The number of epochs is an important factor for improving the generalization ability of CNN to classify better. Class-wise performance increases in our HAM system; thus, overall performance increases as well.

## Conclusion

This thesis is dedicated to study Home Monitoring systems based on acoustic activities. The primary goal was to build a home activity monitoring system and experiment its efficiency. To cope with this, we have supported an experimental case study using a huge dataset from DCASE 2018. Our contributions are categorized as follows:

First, we have examined the effect of combining a feature extraction technique (Log Mel-band Energy) and a classification paradigm (CNN). We have chosen one second segment based F-score for evaluating the performance of our system. From this experimental study, we conclude that the HAM system based on Log Mel-Band energies along with CNN is widely productive in terms of predictive performance with a high overall class-based F1-Score.

Second, we have investigated the effect of increasing the number of epochs on two experiments, namely CNN100 and CNN500 on the generalization ability by performance. Based on our analysis, we can conclude that as the number of epochs increases, the generalization ability of CNN increases as well since the model is trained more times to obtain better generalization. The overall class-based F1-Score was estimated by 77.56% for CNN100 and 84.83% for CNN500.

However, we have noticed that due to the lack of data in this field of research, the amount of extracted features were not sufficient for learning an effective model which influences negatively the performance of CNNs. We believe that a data augmentation technique can lead to a better performance.

### Limitations and Future work

This thesis has invoked several areas in the domain of SED, mainly HAM based on the experimental findings conducted to build a HAM system based on acoustics. A suggested extension of this work would be to exploit data augmentation techniques along with CNNs for a possible improvement in the behavior of such systems in order to reduce overfitting and

generalize better. An other extension is using GPU resources for better performance in terms of execution time.

During this study, we have encountered many struggles. The full dataset was not easy to download, it required a very fast internet flow and a very long period of time. The training of the learning models took a very long time since we trained it using CPU, unfortunately we could not use GPU resources since we were not able to move due to the international pandemic. Thus, storing the trained models caused an enormous increase in the usage of memory space and a considerable period of time.



## Bibliography

- [1] Krstulović, S. (2018). Audio event recognition in the smart home. *In Computational Analysis of Sound Scenes and Events* (pp. 335-371). Springer, Cham.
- [2] Wang, J. C., Lee, H. P., Wang, J. F., & Lin, C. B. (2008). Robust environmental sound recognition for home automation. *In IEEE Transactions on Automation Science and Engineering*, 5(1), 25-31.
- [3] Cakir, E., Heittola, T., Huttunen, H., & Virtanen, T. (2015). Polyphonic sound event detection using multi label deep neural networks. *In 2015 IEEE international joint conference on neural networks (IJCNN)* (pp. 1-7).
- [4] Cakir, E., Parascandolo, G., Heittola, T., Huttunen, H., & Virtanen, T. (2017). Convolutional recurrent neural networks for polyphonic sound event detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(6), 1291-1303.
- [5] Adavanne, S., & Virtanen, T. (2017). A report on sound event detection with different binaural features. arXiv preprint arXiv:1710.02997.
- [6] Lu, R., & Duan, Z. (2017). Bidirectional GRU for sound event detection. *Detection and Classification of Acoustic Scenes and Events*.
- [7] Jeong, I. Y., Lee, S., Han, Y., & Lee, K. (2017). Audio event detection using multiple-input convolutional neural network. *Detection and Classification of Acoustic Scenes and Events (DCASE)*.
- [8] Virtanen, T., Plumbley, M. D., & Ellis, D. (Eds.). (2018). *Computational analysis of sound scenes and events. Springer International Publishing*.
- [9] What is sound?, (2020), oxford languages, Oxford University Press. URL: <https://oxford-googl's language/what is sound?>

- [10] Dekkers, G., Vuegen, L., Waterschoot, T., Vanrumste, B., & Karsmakers, P. (2018). DCASE 2018 Challenge - Task 5: Monitoring of domestic activities based on multi-channel acoustics. *Technical Report*, KU, Leuven.
- [11] Sejdić, E., Djurović, I., Jiang, J. (2009). Time-frequency feature representation using energy concentration: An overview of recent advances, *Digital Signal Processing*, vol. 19, no. 1, pp. 153-183.
- [12] Manolakis, D. G., & Ingle, V. K. (2011). *Applied digital signal processing: Theory and Practice*. Cambridge University Press.
- [13] Shikha, G. (2013). Feature Extraction using MFCC," *Signal & Image Processing: An International Journal (SIPIJ)*" , vol. 4, no. 4.
- [14] Machine Learning and AI via Brain simulations . (2019). Stanford University. Retrieved 2019-08-01.
- [15] Dhanalakshmi, P., Palanivel, S., & Ramalingam, V. (2011a). Classification of audio signals using AANN and GMM. *Applied Soft Computing 11 (1)*, 716–23.
- [16] Babae, E., Anuar, N. B., Abdul Wahab, A. W., Shamshirband, S., & Chronopoulos, A. T. (2017). An overview of audio event detection methods from feature extraction to classification. *Applied Artificial Intelligence*, 31(9-10), 661-714.
- [17] Serizel, R., Bisot, V., Essid, S., & Richard, G. (2017). Acoustic Features for Environmental Sound Analysis.
- [18] Oday, K. H. (2018). *Journal of Information, Communication, and Intelligence Systems (JICIS)*, Volume 4, Issue 5.
- [19] Caka, N. (2015). What are the spectral and temporal features in speech signal?. Retrieved from: [https://www.researchgate.net/post/What are the spectral and temporal features in speech signal?/citation](https://www.researchgate.net/post/What%20are%20the%20spectral%20and%20temporal%20features%20in%20speech%20signal%3F/citation).
- [20] Sehgal, A., & Kehtarnavaz, N. (2018). A convolutional neural network smartphone app for real-time voice activity detection. *IEEE Access*, 6, 9017-9026.

- [21] Peitler, M. (2016). Acoustic event detection of general sounds , Graz University of Technology.
- [22] Fayek, H. (2016). Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between.
- [23] Azencott, C. A. (2018). Introduction au Machine Learning. Dunod. Cambridge, Massachusetts London, England. *ISBN 978-0-262-01243-0*.
- [24] Duda, R. O., Hart, P. E., & Stork, D. G. (2000). Pattern classification. Second edition. N'ju-Jork.
- [25] Bishop, C. M. (2006). Pattern recognition and machine learning. springer.
- [26] Ripley, B. (1996). Pattern Recognition and Neural Networks. Cambridge University Press. p. 354. *ISBN 978-0521717700*.
- [27] Brownlee, J. (2017-07-13). "What is the Difference Between Test and Validation Datasets?". Retrieved 12 October 2017.
- [28] Prechelt, L., & Geneviève, B. O. (2012). Early Stopping — But When?. *In Grégoire Montavon; Klaus-Robert Müller (eds.)*. Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science. Springer Berlin Heidelberg. pp. 53–67.
- [29] Parascandolo, G. (2015). Recurrent Neural Networks For Polyphonic Sound Event Detection. Tempere university of technology, Tempere, USA.
- [30] Matusugu, M., Mori, K., Mitari, Y., & Kaneda, Y. (2013). Subject independent facial expression recognition with robust face detection using a convolutional neural network, vol. 16, n° 5, 2003, p. 555–559.
- [31] Rojek, I., Jagodziński, M. et al. (2012). Hybrid artificial intelligence system in constraint based scheduling of integrated manufacturing ERP systems. *In Hybrid artificial intelligent systems, Eds. Corchado, E., Snášel, V., & Abraham, A. Springer Berlin Heidelberg, 7209: 229–240*.
- [32] Dertat, A. (2017). Applied Deep Learning - Part 4: Convolutional Neural Networks.

- [33] Zhou, X., Zhuang, X., Liu, M., Tang, H., Hasegawa-Johnson, M., & Huang, T. (2008). HMM-Based Acoustic Event Detection with AdaBoost Feature Selection. *In International Evaluation Workshop on Classification of Events, Activities and Relationships*, pp. 345–353.
- [34] Chew, J., Sun, Y., Jayasinghe, L., & Yuen, C. (2018). *Engineering Product Development*, Singapore University of Technology and Design, Singapore.
- [35] Nakadai, K., & Danilo R. O. (2018). Research Div. Honda Research Institute Japan, Wako, Japan.
- [36] Alon R., & Alon, A. (2018). Signal Processing Department, National Research Center, Haifa, *IEE, Technion*.
- [37] Dekkers, G., Lauwereins, S., Thoen, B., Weldegebreal Adhana, M., Brouckxon, H., Waterschoot, T., Vanrumste, B., Verhelst, M. & Karsmakers, P. (2017). The SINS database for detection of daily activities in a home environment using an acoustic sensor network. *In Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE2017)*, 32–36.
- [38] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., & Nieto, O. (2015). Librosa: Audio and music signal analysis in python. *In Proceedings of the 14th python in science conference (Vol. 8)*.
- [39] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016). Tensorflow: A system for large-scale machine learning. *In 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)* (pp. 265-283).
- [40] Carneiro, T., Nóbrega, R. V. M., Nepomuceno, Da., Thiago, B., Albuquerque G. D, Filho, M., & Rebouças, P.P. (2018). Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications, *IEEE*.
- [41] Cunningham, T. (2015). Python, pickle Security problems and solutions.
- [42] Alpaydin, E. (2009). *Introduction to machine learning*. MIT press.
- [43] Moudiki, T. (2020). Linear model, xgboost and randomForest cross-validation using crossval:crossval\_ml. *In R-bloggers: R news and tutorials contributed by hundreds of bloggers*.