uc3m | Universidad Carlos III de Madrid

University Degree in Audiovisual Systems
Academic Year (e.g. 2014-2019)

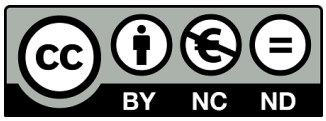
Bachelor Thesis

"Violent event detection from acoustic signals"

Óscar Otero Martínez

Carmen Peláez Moreno

Madrid, February 8, 2020



SUMMARY

Keywords:

DEDICATION

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Context

Violence against women remains an invisible phenomenon, deeply within the victim's private life in most cases. It is based on deep social and cultural roots and it is undoubtedly linked to unbalanced relationships between men and women in different situations and contexts, such as economics, politics and religion. In order to prevent these conflicts, the related legislation has achieved important improvements for the last years. According to the results of most of the studies, victims can be usually defined as women who endured violence during their childhood and felt socially isolated. They are also characterized for a considerable economic dependency and a low educational level.

With the purpose of making a difference when identifying situations showing this kind of violence and apply all the knowledge and technological advances acquired during this information era, machine learning and deep learning models can collect all the available data to protect eventual victims.

The main goal is to get to know how the victim is feeling, for example, if she is scared or nervous, and combine this with other variables which may play an important role in the scene and might be helpful in making a decision about the characterization of the ambiance. There are several factors that can be considered to achieve this task. One of them is the audio, either the victim's voice or the environmental sounds.

Plenty of useful information can be extracted from the acoustic scene of a certain place. The detection of audio events is an equally good way to define what is happening in a certain moment. Once these data are collected, they can be classified in different categories and thus describe the scene. Based either on an objective definition of gender violence or in an explanation previously obtained from a particular/specific victim, this acoustic knowledge can be interpreted as dangerous for the user.

1.2. Objectives

The utilization of learning models to extract useful information from the worlds data has become a very common practice in most of the fields. One type of habits that have gained a lot of popularity in the scientific community is the use of multimedia data. In many cases, the samples used to train the models consist in images that belong to a certain kind of problem, such as medical imaging or object recognition. This field is known as Computer Vision (CV). Many world well known architectures and enormous data bases have been born during the study of this kind of problems.

In the same way, audio data have been used to get conclusions from a lot of real

world problems. In order to tackle the task of violent event detection it is important to decide what perspective is going to be taken into account when defining a violent event, whether an objective point of view or a more personalized standpoint according to the victim criteria. Apart from this, it is also necessary to extract the required features, that is, information from the audio signals that will allow to train the models so to get the results. However, the main work will be characterized by classifying a whole scene depending on the events this is built by. Once an action sound is categorized, it can be identified as violent by checking if it belongs to the violence definition previously defined.

The different acoustic scenes that may be considered for the problem can be composed by events of different nature or those that belong to just one class. This difference may cause that the techniques utilized to address the problem can differ. As a further approach, it is interesting to find a method that can distinguish among events that come from different sources of audio.

1.3. Regulatory framework

Tips?

1.4. Socio-economic environment

Tips?

2. STATE-OF-THE-ART

Since the main topic of this work has not been treated in the same way before, we decided to start our search in the literature addressing two topics that have been studied largely for the last years related to acoustics scenes and events.

2.1. Acoustic Scene Classification and Acoustic Event Detection and Classification

Acoustic Scene Classification (ASC) refers to the association of an audio sequence to a certain semantic label that describes the environment in which it took place [1]. With this idea in mind, the classification of acoustic sceneries has been tackled with two different kinds of concepts: soundscape cognition, **or** understanding how the human being perceives the sounds subjectively from the physical environment that surrounds them [2], and Computational Acoustic Scenes Analysis (CASA) **or** working on new computational methods that may help automatize this task through machine learning and processing signal techniques [3]. This notion can have many applications, such as content recognition - by allowing devices to obtain benefits and information from its situation [4], for medical utilizations [5], as a tool for musical recognition [6] or for a complement to Computer Vision (CV).

Simultaneously to these advances in the ASC, another related area has evolved during the last years. Some computational work has been deployed for the tasks of Acoustic Event Detection and Classification (AED/C). It can be described as the processing or treatment of sound signals in order to convert them into significant descriptions that match a listener's sensing of the events and sources composing the acoustic environment [7]. The detection part consists on identifying the events in a temporal stream of audio and labelling them. The result is usually accompanied by the time interval in which the occurrence is set. However, the classification is a task that acts directly on the event that has been already isolated and has the purpose of designating a label or class to the sound [8]. These techniques have had plenty of applications, e.g., in the medical field [9], in biological topics such as bird noise detection [10], and for multimedia information retrieval from video sources in social media [11].

2.1.1. Features and methods

As in many fields, the frontier between features and methods used for audio tasks have been blurred during the last decades. Usually, a problem is addressed with a pre processing period of the data and then the model or method is implemented. Right now, these two stages are sometimes maintained but also have been mixed or changed depending on how the algorithm used works. In this chapter, we are going to try to explain these difference

between features and methods and try to separate them in order to ease its understanding.

Features

In every machine learning or pattern recognition task, for the system to be able to infer and extract conclusions from the input given, it is necessary a pre processing period in order to make some transformations to the data so this can be readable by the model. This stage is known as feature extraction and the goal is to convert the original information into a set of of values or vectors that characterize the data regarding some desired properties [12].

There are several ways that allow us to perform this processing stage. The most common and basic one consists on extracting features that are really closely related to the original signal which are called low-level descriptors (LLD) [13]. These are computed by performing some mathematical operations or formulas to the original data that can be considered rudimentary when comparing with other techniques. However, they are really extended and still in use nowadays [14].

In the audio field, there are two types in which all LLD can be grouped into. One of them is for the features that have been computed by considering the audio signal in its original form in the recording, i.e., in time-domain, and they are known, of course, as *time-domain audio features*. The other case refers to those characteristics that are obtained from the signal after been transformed to frequency domain. These are commonly known as frequency-domain or spectral audio features. For the procedure of feature extraction, the signal is usually divided into frames that can be overlapped by using a sliding window so the calculations are done per frame, obtaining a final matrix with size $number\ of\ frames \times number\ of\ features$ [12]. It must be taken into account that the final application of the whole system is going to completely influence in which features have to be computed. For example, not the same features are used for speech recognition than for musical information retrieval. In tables **??**, **??** a summary of most used time and spectral features is included respectively.

2.1.2. Features and methods

In the literature, numerous articles have been published related to ASC field. These can be sorted into two different currents based on how the problem is addressed. One of them considers the scene as a single instance with the purpose of representing it through a long-term statistical distribution that models a set of low-level features [15]. An acoustic event can be characterized in different ways for this type of method. In previous works, some of the common habits usually used in speech recognition tasks played a main role in the extraction of features, such as the fundamental frequency (F0), the F0 envelope and the probability of voicing. Apart from these, also spectral features, as Mel-Spectrum bins, zero crossing rate (ZCR) and spectral flux (SF), and energy features, such as the energy in

Feature	Description
Energy Entropy	It is a useful to detect sudden changes from the energy of a signal. To calculate this value for a certain subframe, it is necessary to first compute the normalized energy of the subframe with respect to all the frames energy [29].
Short time energy	It is the energy for a short segment of signal. It is normally used in speech tasks in order to identify voiced form non-voiced fragments [33].
zero crossing rate (ZCR)	This can be defined as the number of times the amplitude of the signal crosses the zero line, i.e., changes from negative to positive. It is computed by the number of zero-crossings by the amount of samples in the frame [29].

Table 2.1. TIME-DOMAIN AUDIO FEATURES

Feature	Description
Spectral Flux	It is computed to measure the spectral changes between two successive frames. To do so, the difference is calculated through their squared spectra coefficients normalized [29].
Spectral Rollof	This represents the skewness of the shape of the spectrum by given the frequency below which a concrete percentage of the magnitude distribution of the frequency transform is concentrated [33].
MFCC	It is a feature that it is commonly used in speech recognition because interprets the frequency bands in a very similar way to human perception. It is computed from the STFT. **A detailed explanation can be found in appendix ??** [33].

Table 2.2. SPECTRAL AUDIO FEATURES

bands or the logarithmic-energy [16] had an important function on this task. However, the best results have been achieved with the so-called Mel-frequency Cepstrum Coefficients (MFCC), defined as a cepstral feature, **which will be explained further on**. This kind of characteristics extracted from the audio can be called low-level descriptors and they are usually combined with algorithms and methods to address the classification task. In this "bag-of-frames" approach, in which the scene is considered as a single object, a typical technique was to model the samples features into global statistical characteristics from

Explain more MFCC?

the local descriptors by using Gaussian Mixture Models (GMM) [17].

There is another path to dig for Acoustic Scene Classification, which consists on including a representation of data prior to the classification, which transforms the scene by using a set of high level features normally obtained with a vocabulary or dictionary formed by acoustic atoms. These are usually a depiction of events or streams within the scene and do not need to be known a priori [15]. Apart from the typical well-known audio features, the ones named above as low-level descriptors, there exists other acoustic characteristics which may seem to be hidden in the data but can be found by using unsupervised-learning methods. This is the way to act when dealing with the above mentioned acoustic atoms.

One of the approaches that can be found in the literature about this idea is based on the use of a previously learned overcomplete dictionary that is utilized to sparsely decomposed the spectrogram of audio. This dictionary will be used by an encoder with the purpose of mapping new input data to real similar versions of their own sparse representation in a fast and efficient way. Finally, the obtained codes will feed a Support Vector Machine (SVM) classifier, used for the task of music genre prediction [18].

Include results?

Another job done in the sparse-feature representation framework presents a way of mixing high feature learning techniques with a pooling method for the objective of music information retrieval and annotation. After some preprocessing of the audio signals data, three feature-learning algorithms are trained finding that sparse restricted Boltzmann machine (sparse-RBM) gets better results than K-means and Sparse Coding. Once the features are obtained, an extra step takes place before performing the classification task, the one called pooling and aggregation. The goal of this procedure is to achieve a feature representation for a long sequence such as a song. Since when joining short-term features that belong to small segments inside the song may result in a loss of their local meaning, a max-pooling operation is computed over each subsegment in order to just consider the maximum value for each feature dimension. After that, these are aggregated by computing the average. The max-pooling contribution resides on reducing the smoothing effect when averaging the values [19]. This approach is feasible because of the homogeneity in music data. However, this technique could be slightly risky when dealing with acoustic scenes. For this case, a modified version of this method has been proposed. Taking into account that the presence of events is less frequent, instead of considering the whole long sequence to apply the max-pooling for, it will just be used in those segments that had been already detected as significant events by establishing a threshold value and setting an onset and offset that allow to know the start and end time [20]. **The representation of the audio event in a feature space explained in this case is the one that better fits our approach (3.3) until now.**

Include picture of the pipeline?

The classification of acoustic scenes can be linked to event detection. The working method used for this task is really similar to the one used for ASC. Thus, it is not surprising that most of the works found in the literature address this task with the use of MFCC as features and with techniques such as HMM or GMM. For the purpose of find-

ing the desired events, the whole detection process can be split into two parts. Firstly, a classification of already isolated events should be executed in order to build a vocabulary of acoustic actions. In this case, the data used belong to short-term sequences that must strongly show the semantic meaning of the corresponding event. This is important because there may be more acoustic representations in the same short segment than the one to be detected, but this must stand out among the others. Consecutively, for the detection part, the input data will be composed by long tracks so that time allocation of the events will be implemented. Therefore, after obtaining the different short segments from dividing the long sequence up, they will be classified considering the results from the first step [21].

Go deeper? HMM, GMM, MFCC

A novel type of feature have been used in the last years that differs from the already explained low-level and high-level. Deep Neural Networks (DNN) have grown increasingly for plenty of classification tasks and so in the multimedia area. The problem with these type of systems is the huge amount of data that is needed to make them work properly, which can be translated in a lack of labelled data. One of the habits that has been currently resorted by the researchers consists of learning what is called deep data *embeddings* from extensive collections of, ir our case, audio and use them so as to perform **shallow classifications by using simpler datasets**. There have been implemented some models about this topic, such as Look, Listen and Learn (L^3) [22] net that uses as input for the the embedding extractor the linear-frequency log-magnitude spectrogram of 60 million audio samples, the system called SoundNet [23] that has been designed to obtain embeddings from training a deep audio classifier in order to predict the output o a deep image classifier and the VGGish network, designed by Google researchers. This last case is the one we used in this work and it will be explained in more detail in section 3.2.

2.1.3. Transfer learning

All these models previously mentioned allow to use complex extractors trained with huge collections of data and apply them to models considerably much less complicated to address other type of problems. This is possible due to a kind of techniques commonly known as transfer learning.

The typical consideration in plenty of machine learning tasks consists on extracting the training and testing subsets of data from the same feature space and same distribution. When one of these initial assumptions change, it is necessary to rebuild the whole model from the initial point, including new training data, which means a lot of computational cost and loss of efficiency [24]. Its working manner can be explained as an analogy of how humans transforms their ability on a certain task to obtain knowledge for other purpose. An example could be how musicians apply their previous experience to get to know faster how to play other instrument.

The first work in which this topic has been treated widely was in 1995 in the workshop *Learning to Learn* [25] and since then many approaches have arisen and baptised

the same idea with different names such as knowledge consolidation or inductive transfer . However, it was 10 ten years later, in 2005, when the first idea of the ability of a system to identify and apply learned skills previously to completely new problems appeared from the hand of the Broad Agency Announcement (BAA) 05-29 of Defense Advanced Research Projects Agency (DARPA)'s Information Processing Technology Office (IPTO) [24]. This can be expressed as a relation between a *source* task, where the abilities are learned, and *target* task, the novel problem that needs to be resolved. As a difference with other similar methods, in this concept of transfer learning the roles of these two are not equal since the weight of the target is much heavier. In figure 2.1 it is shown the difference between a common machine learning approach and the use of transfer learning.

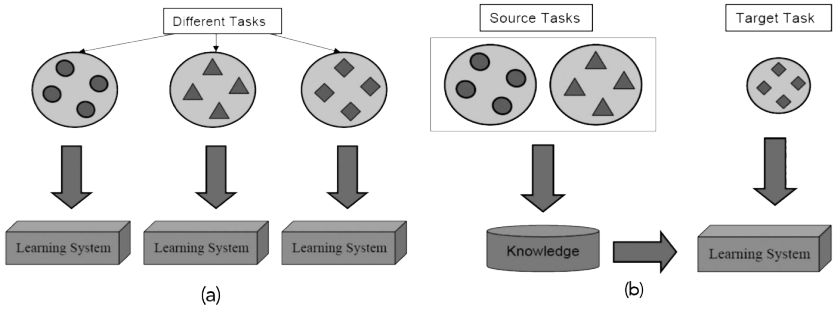


Fig. 2.1. Difference between traditional machine learning (a) process and feature learning (b) [24]

The same idea can be understood from a mathematical vision that analyses the relation between the two different spaces from types of targets [24].

Considering a *domain* D that is composed by a feature space denoted by X and a marginal probability named $P(X)$, where $X = \{x_1, ..., x_n\} \in \chi$. The whole domain can be expressed as $D = \{\chi P(X)\}$ where $x_i th$ is a certain vector inside the feature space.

In the same way, a *task* can be defined as T formed by a label space γ and an objective predictive function η. The task formulation is $T = \{\gamma, \eta\}\}$. This predictive function cannot be observe, however the intention is to learn ir from the training data, that is composed by pairs of the form $\{x_i, y_i\}$, $x_i \in X$ and $y_i \in \gamma$.

The predictive function η can be used to predict a corresponding label of a new sample x. From a probabilistic perspective, this new label can be expressed as $P(y|x)$. So, the task T can be defined as $T = \{y, P(Y|X)\}$, in which $Y = \{y_1, ..., y_n\} \in \gamma$. For each vector x_i, the function η finds a prediction y_i.

Once these parameters have been defined, considering the source domain D_S, task of source domain T_s, the target domain as D_T and its respective task as T_T, the transfer learning has the purpose of obtain the condition distribution in the target domain $P(Y_T|X_T)$ with the information extracted from D_S and T_S where $D_S \neq D_T$ or $T_S \neq T_T$ [26].

2.2. Violent Event Detection

All the multimedia information available can be applied to many fields and in different connotations. One of the slopes that has appeared in the acoustic scenes and events sphere is the one applied to violence. For this case, an essential point before addressing any problem is to decide what kind of definition the word violence is going to adopt since it is a really subjective concept. An objective perspective has been given by the World Health Organization as "The intentional use of physical force or power, threatened or actual, against oneself, another person, or against a group or community, that either results in or has a high likelihood of resulting in injury, death, psychological harm, maldevelopment or deprivation" [27]. There are other definitions found in different works as "physical violence or accident resulting in human injury or pain" [28] or "any situation or action that may cause physical or mental harm to one or more persons" [29].

Recent studies have treated this problem in different ways due to all types of conditions that this may take place in. During the last years, the possibility of creating and providing audiovisual content has grown widely, which has led to an enormous variety of topics in which, some of them, could be considered unappropriated for certain parts of the audience. This is the reason why there have just been done works related to the field of video content analysis and detection of violence. In some cases, audio and image features have been combined to address these problem [30]. However, it has been found that sound information could be really useful and a more efficient way of working compared to image, since it is easier to process and the cost is lower. Related works have utilized audio features in the time-domain and in the frequency-domain, similar to the ones explained for ASC, then combined with a normal SVM classifier [29]. Other researches have tried more complicated models with the intention of improving the classification task. It is the case of using DNN, fed with both image and audio data, which performs the task more efficiently [31]. Violence detection has also been used for other applications such as video surveillance. For example, one of the scenarios for this purpose consists on preventing violent acts inside elevators [32]. For this case, the considered dangerous situations are composed of anti-social actions that are likely to happen in this kind of places, concretely, urinating, vandalism and attacks on vulnerable victims, such as women, children or elderly. The framework proposed is based on audio-visual data, but the master classifier is driven by audio, due to the possible subtleness of the scenes that are desired to detect. So, first the audio incident detector triggers the process when a non-silent event takes place. Then, the image processing begins in order to extract information related to who is involved in the action and how aggressive is it. Another utilization of the surveillance approach is its use for the evolution of smart cities [33]. For this goal, since the system will be implemented in real-life environments, one of the advantages about working with data coming from sounds is the respect for privacy, that, otherwise, using video recordings it would be violated.

The difference in these two applications, apart from the task they are addressing,

resides on the data they are working with. For violent content analysis, the data usually comes from fictional audio sources as movies or video-games. However, for real-environment systems, the data is extracted straight from actual day-to-day life situations. In this second case, some disadvantages can be appreciated. For example, the signals are not preprocessed, which means the original properties of the sound are not modified so the processing part before classification becomes tougher. Also, the presence of background noise is more common and loudness of some events, as speech, may vary with time [34].

Makes sense?

2.2.1. Gender-based violence

Throughout history, women have been an object of abuse and suffering in many different situations even though in those that were considered as their familiar surroundings. They have been bashed, sexually harmed and psychologically maltreated by those who were supposed to be one of their closest intimates [35]. In the same way, in the recent times, late studies have shown that 35% of women from all over the world have been victims of physical or sexual damage [36], and 43% of women from Europe have declared going through some psychological or mental violence at least once in their lives [37]. In this context, it is necessary to define the concept of gender-based violence, which can be described as the multitude of harmful behaviours that are focused on women and girls just because of their sex, such as female children and wife abuse, sexual assault, dowry-related murder and marital rape, among others. Particularly, violence against women involves any act of verbal or physical force, extortion or lethal denial which has a woman or girl as a target and provokes the physical or psychological hurt, humiliation or irrational privation of liberty and contributes to continue women subordination [38]. Within this definition, it can be considered that most of the times that these violent situations take place, they are originated due to persons that are supposed to be part of the victims' closest circle of trust, i.e., their husbands or boyfriends. This is called Intimate Partner Violence (IPV) and it is recognized as a public health problem affecting women across their life span resulting in different undesirable unhealthy outcomes, such as depression, chronic pain and even dead [39].

Glossary?

2.2.2. Our point of view

Explicación EMPATIA?

As a contribution to the EMPATIA-TC project developed by Universidad Carlos III de Madrid, the main goal in this work is to make progress in detecting gender-based violence situations, specifically applied to day-to-day scenes, in which IPV is likely to be present. One of the parts from the proposed system is composed by wearable devices that the victim can carry to collect diverse types of information and process them to obtain conclusions and increase the efficiency. Among these accessories, we can find a pendant that pays attention to the user's voice and the surrounding audio to analyse what is happening at a certain moment. For our purpose, the interesting part resides on achieving auditory data so as to detect violent incidents that are formed by sounds already known

for characterizing these episodes considered dangerous by the victim.

The definition that is assigned to violence is really important in order to define which audio events should be taken into account. However, considering the subjectiveness of this concept, categorizing violence for every type of user is an extremely difficult task. For this reason, the final idea to answer this question is to make the victim able to decide which kind of hearing events the system must be aware of. In the complete project, this can be carried out by a phone user interface which displays a list of sound events and she has the labour of picking up those that are violent according to her criteria. Since the development of this tool is out of the scope of this work, we have decided to implement a simpler mechanism which will be explained in subsection 3.3.1.

2.3. Databases

A fundamental objective was to find a database that allows for building a system with the desired characteristics, so a rich variety of acoustic events is needed with an essential big representation of violent sounds. In the table 2.3 is represented a relation of the different databases that have been considered for the realization of this work.

The last three options shown in table 2.3 are the ones that better adapted to the problem of the work. *VSD Benchmark* was the first option we were happy with. Within the two ways of working given, the movies and the YouTube videos, the former was the easiest to use since the annotation specified exactly what kind of violent events were present in the scene and the onset and offset within the whole film. However, to access the data it was necessary to pay. The latter was alright but it just indicated the presence of violence, without determining the type of event. Another choice was *Freesound dataset* because it is formed with all type of videos so we could extract those classes that are more interesting for us. However, it is still in an annotation process and it is not ready to download yet. As a final conclusion, we decided to go for *AudioSet*, which **will be explained further on**.

Name	Description	Considerations
URBAN-SED [40]	10,000 soundscapes with sound events. Every soundscape contains 1 to 9 sound events with strong annotations.	Events are completely specified but it just contains three interesting types of classes.
UPC-TALP [41]	It belongs to CHIL project, for the AED task. Isolated acoustic events that occur in a meeting room environment.	Payment is needed to achieve the data and the classes are a little out of our topic.
MIVIA: Audio Events Data Set for Surveillance Applications [42]	6,000 events with background noise.	The classes included belong to our topic, but they are just three: glass breaking, gun shots and screams.
TUT rare sound events [43]	Source files for creating mixtures of rare sound events (classes baby cry, gun shot, glass break) with background audio.	Similar problem to MIVIA: just from three interesting classes.
IEEE AASP Challenge [44]	Composed by ASC and AED. It is formed by two subtasks: OL (Office-live) and OS (Office Synthetic)	Labels for both subtasks are out of our scope since they are likely to happen in an office environment: keyboard clicks, hitting table, etc.
TUT-SED Synthetic 2016 [45]	Isolated sound event samples were selected from commercial sound effects	The variety of classes is large enough but for our purpose just four of them are useful.
VSD benchmark [46]	Violent events from 32 Hollywood movies and 86 YouTube web videos, together with high-level audio and video concepts.	Payment is needed to purchase the movies and the videos do not specify the type of violent event
AudioSet [47]	An ontology of 632 audio event classes and a collection of 2,084,320 human labeled 10-seconds sound clips from YouTube videos.	The final pick. Plenty of the videos have more than one audio label but we were able to adapt the data to the problem because of the huge amount of clips.
Freesound dataset (FSD) [48]	Filling AudioSet ontology with 297,144 audio samples from Freesound.	This may seem a very good option as well but it is not available yet.

Table 2.3. TABLE OF STUDIED DATABASES

3. METHODS

3.1. Database: AudioSet

3.1.1. What is AudioSet

How cite?

Audio Set can be described as a sound large-scale dataset that has the intention of putting the availability of audio and image data on the same level. It is composed by a huge variety of manually-annotated audio events and is organized by following an important ontology formed by 632 different audio classes. The data has been extracted from YouTube videos and the labelling process has been based on diverse factors such as metadata, context and content analysis. It has been developed by Google with the purpose of producing an audio event recognizer that can be applied to plenty of acoustic situations coming from the real world [47].

3.1.2. Ontology

In order to put this dataset together the events have been organized in an abstract hierarchy. This is composed by higher-level classes which describe a certain type of sound and also acts as parents of other labels that refer to more specific events. With this purpose, the relationship among different classes needed to be non-exclusive, so labelling similar audio events may result into a more general class, the parent, if there existed ambiguity. This is also helpful for labellers due to group the clips in an easier and faster way.

The Audio Set Ontology has been made considering some fundamental guidelines as the ones explained below:

- **A complete collection of all labels must be prepared** so that it can be used to define sound events from real-world aural data.
- When labelling an audio event the result must match the criteria of a common listener.
- Different categories should be easy to distinguish by an ordinary listener. In the case that two different labels do not satisfy this requirement, these should be merged. With this condition, the spectrum of possible labels remains limited.
- The distinction of two different classes must be done by relying just on the audio, it cannot be accompanied by image or visual information.
- The hierarchy should not be very deep by keeping the number of children per parent class to no more than 10. This also eases the annotation labour.

It is easy that an ontology of this volume gets leaned or biased in a particular direction due to several factors, such as the subjectiveness of its creators or the selection of the initial set of classes used when starting to work. With the intention of generating a primary list that covers a wide range of audio events in an objective way, the researchers decided to apply an impartial, web-scale text analysis from the very beginning. They agreed on detecting hyponyms of the word "sound" by utilizing a modified version of the famous technique called *Hearst patterns* [49], a method proposed to automatically acquire hyponymy lexical relations from unrestricted text. As a result, an enormous collection of terms came up. This was filtered by considering how well these terms represented audio, i.e. by combining together the global frequency of occurrence and how exclusively these are recognized as hyponyms of "sound" instead of other terms. As an output of the final process, a list of 3000 terms was obtained.

With respect to the hierarchical relation among categories, it was constructed by the authors with the main intention that this satisfied their human comprehension of the sounds. Event though this is a subjective manner, it also makes sense since this is how the hierarchy best performs its labour on helping human labelling. After all the organization process, the model is not based on a strict behaviour as a singular node can appear in many different locations, i.e., a single node can be child of different parent nodes. The final result is composed by 632 audio event labels and, in the hierarchy, there is no deeper case than 6 levels. Figure 3.1 shows the nodes that belong to the two top layers of the ontology.

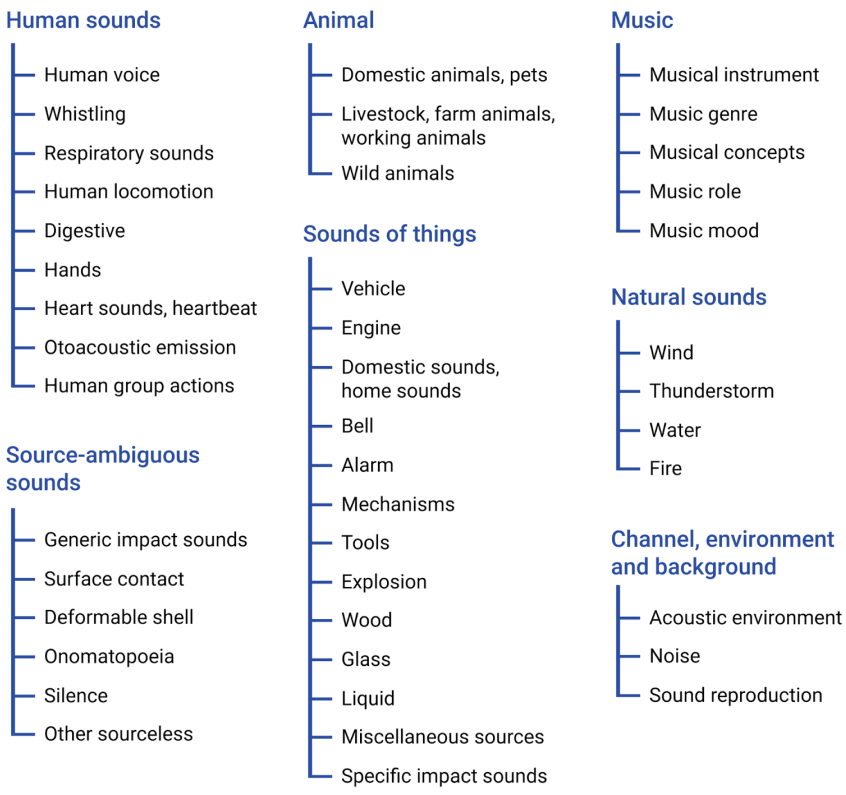


Fig. 3.1. First two layers of Audio Set Ontology [47]

This whole structure has been given to the user as a file in JSON format. A couple of fields have been included for each label in order to describe its meaning and make clear its position within the hierarchy. A description for all of these can be found in table 3.1.

ID	This field includes the Knowledge Graph Machine ID (MID) that best describes the sound or its source. It is used as a primary identifier for the class.
Display name	Short name formed by one or two words that identifies the audio class. It sometimes includes a small alternative separated by a comma so it does not feel ambiguous.
Description	One or two explaining sentences so the meaning of the category is more defined. These can be extracted from Wikipedia or WordNet.
Examples	At least one example of the label is provided as a URL of a YouTube video.
Children	An array filled by the Machine ID (MID)s from all immediate children of the class.
Restrictions	It specifies if the category in question either has been discarded or there are no audio clips under it.

Table 3.1. FIELDS PER CATEGORY IN THE ONTOLOGY

For the field *ID*, the identifiers are known as Machine ID (MID) and belong to the *Knowledge Graph* designed by Google [50]. This is a knowledge base that Google services use to improve the quality of its search results and it is composed with information extracted from a wide variety of sources. The MIDs are the identifiers of the different elements that belong to this huge dictionary. For instance, the MID of the word "Speech" is "/m/09x0r". Another field that deserves a special explanation is the one corresponding to *Restrictions*. Within all the categories of sound events, there are two flags that indicate an exclusive behaviour that differ from a typical label: "blacklisted" and "abstract". The former refers to a class that has been hidden from labellers due to its confusing meaning. The latter has been used for those classes that are just utilized as intermediate nodes in order to provide a better grouping inside the organization, and are not expected to be used in the implementation tasks. In total, out of the 632 categories, 56 have been categorized as "blacklisted" and 22, as "abstract".

3.1.3. Data

The different YouTube videos that constitute the dataset are included in a .csv file in which each row is formed by the video identifier, the start and end time of the audio event within the video and the ontology labels that the certain clip belongs to. All the video segments

have a longitude of 10 seconds maximum, except from those that the original video is shorter, then the whole thing will be considered as the an audio event.

The final release of the dataset is composed by 1,789,621 segments, with a duration of 4,971 hours of video and audio. After executing the selection process in which the different labels were populated with the final corresponding segments, a total of 527 classes were gathered, out of which 485 counted with at least 100 samples.

Include any explanation about ratings?

3.1.4. Data access

The data can be obtained through the website [51] in two different formats:

- Files in .csv format that include for each video segment its YouTube video ID, start time, end time and the one or more labels it belongs to.
- Instead of the audio files themselves, they provide already extracted audio features for each segment in compressed files that can be easily downloaded.

For our purpose, we worked with the dataset in both different ways. However, we got further with the already extracted features. These are obtained by using a model called *VGGish* [52] which has been pre trained on a preliminary version of the database YouTube-8M [53]. The features are available to the user in TensorFlow (TF) [54] record files and the code for the VGGish model is also included in a public repository.

include a type of introduction for vggish?

3.2. Feature extractor

In order to best describe the just mentioned VGGish mode, it is necessary to first explain the original network it was based in.

3.2.1. Visual Geometry Group (VGG) model

Convolutional Neural Networks (CNN) are usually able to achieve really good results and even improve human skills on Computer Vision tasks, for example, on recognizing object in an image. With the exponentially growth of the researching works about this field, some challenges have appeared so as to promote the creation of new systems and test their efficiency and results. This is the case of the ImageNew Large Scale Visual Recognition Challenge (ILSVRC), based on the database of the same name, ImageNet. As a solution for the proposed exercise, the investigators from the Visual Geometry Group (VGG) in the University of Oxford implemented a new system achieving the first position and winning the challenge in 2014 [55]. The work they proposed consists of a study of the depth in a Convolutional Network architecture and how this can affect to the accuracy on the goal of large-scale image recognition [56]. To try this, it was necessary to increase the

Appendix to CNN?

number of layers in the network, which was viable due to use a small size of convolutional filters in all of them.

For the training step of their system, they used an input image with standard size of 224×224 in RGB format. The principles to build the architecture are detailed below:

- The input image crosses a bunch of convolutional layers in which the kernel has a size of 3×3.
- The convolutional stride has a value of 1 pixel.
- The padding is fixed to 1 pixel, so the dimensions of the input do not change during the convolution.
- Max-pooling is also included with a window size of 2×2 and a stride value of 2 pixels.
- Two Fully-Connected (FC) layers with 4096 channels after all the Convolutional Network layers.
- One FC layer with 1000 channels to perform the ILSVRC classification.
- Soft-max layer for the final layer
- All hidden Convolutional Network layers use the non-linear function ReLU as **activation function**.

Normalization

All the designs that the creators came up with are based on these initial guidelines, except from just one case where Local Response Normalisation (LRN) is applied.They just differ from each other on the number of layers, starting with 11 the first approach and ending with 19 the last one. The different architectures are specified in the table 3.2 and are ordered from A to E.

3.2.2. VGGish model

The model we used in our task presents a configuration with principles similar to the ones explained in the previous subsection 3.2.1 but with slightly changes that the developers have included in order to adapt it to the audio approach.

The architecture is based in the configuration A from table 3.2 with 11 weights. The differences respect to the original network are listed below:

- The input size was change from 224×224 to 96×64 because of the log-mel spectrogram audio inputs.
- They built the implementation with just four groups of convolutional and max-pool layers so the fifth one was dropped.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 3.2. VGG CONVNET CONFIGURATIONS

- For the last FC layer, they decided to build it with just 128 channels since it is the one that compacts the final embedding.
- Also the *Softmax* layer is not used.

In figure 3.2 is shown how looks the configuration of the final VGGish model.

Input stage

Before passing the data through the whole CNN, a preprocessing stage have been included by the developers in which the input audio will suffer some transformations.

1. In first place, after loading the input audio, the sample rate and the number of channels are checked to be 16 kHz and monochannel, otherwise the file is transformed to satisfy these conditions.

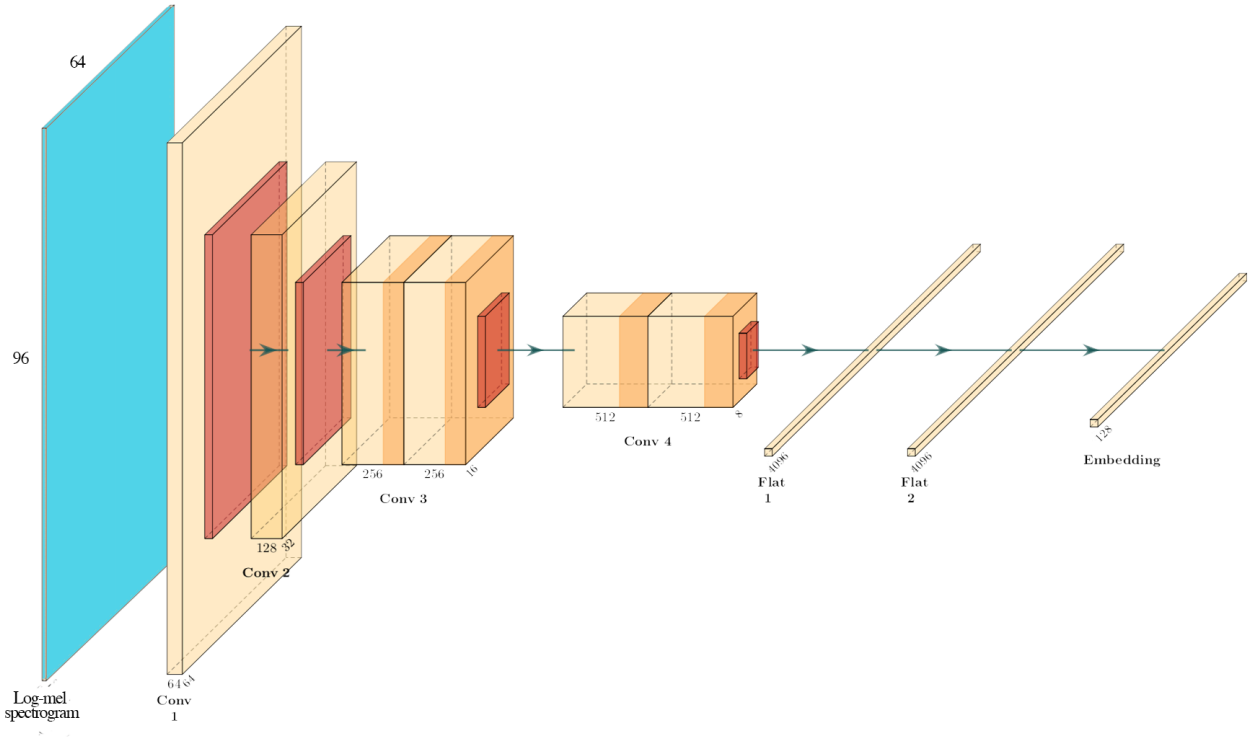


Fig. 3.2. VGGish architecture

2. When the data is ready, the next step is computing the spectrogram. For this, it is necessary to calculate first the Short-Time Fourier Transform (STFT) of the signal. The operation is performed by using a sliding window of 25 ms of type Hann with a hop size of 10 ms. As a result, just the magnitudes that correspond to the positive frequencies are taken, since the ones corresponding to the negative part of the spectrogram are the conjugate of the others.

3. Transforming the spectrogram to Mel scale is what follows. To do so, they compute all mel frequency bins that are going to modify the values of the spectrogram in frequency domain by following a hertz to mel conversion formula . The result is a citation? mel spectrogram from 125 Hz to 7,500 Hz divided in 64 bins.

4. Then, the log-mel spectrogram is calculated by doing the logarithm of the previous result plus a small offset value **to avoid the logarithm of** 0.

5. As a final step, they compute a framing operation over the log-mel spectrogram. The resulting are non-overlapping examples of 0.96 seconds, in which 64 mel bands and 96 frames are contained, each frame with a duration of 10 ms.

Therefore, the result obtained is an ensemble of 10 frames, approximately one per second, each of them with size 96×64, i.e, 96 frames and 64 mel bands.

Embedding stage

Once the initial processing part is done and the log-mel spectrogram matrix is computed and divided into the desired number of frames, it is used as input data for the VGGish CNN. After all the computations inside the network, each example is converted to an embedding of size 128 giving a result of one of this per second of the original audio file. It is good to mention that in this part, a pre trained checkpoint file is loaded.

Postprocessing stage

As final step, they performed some processing of the resulting embeddings. A Principal Components Analysis (PCA) transformation is done joint with a whitening process. Also, a quantization to 8 bit for each embedding element. All these actions are computed with the purpose of making the final output compatible with the embeddings obtained from the YouTube-8M database.

Transfer learning and why choosing this type

3.2.3. Why VGGish

For our goal, one of the toughest tasks consisted on the selection of data that properly adapted to our problem and its preparation so as to obtain features that allowed us to characterize every acoustic event from a violent point of view. Since our first efforts of finding an available dataset characterized for being rich in violent scenes were driving us to a dead end, we decided to take advantage of a huge database which let us rethink the standpoint about how we were going to address the problem. As it was mentioned in subsection 2.2.2, one of the main questions was how to define the term violence for each victim depending on how her certain situation. After finding the Audio Set database, previously explained in section 3.1, with all its variety of samples, we had a wide range of audio data to work with. This is how we came up with the system explained below in 3.3.1.

At this point, not only we had an idea but we had already found a data resource to start with. However, the issue was related to what kind of features could be extracted in order to categorize events from different nature with a unique violent label. An acceptable conception of the term violence could be expected to cover all kind of short high gain events such as hits, smashes, gunshots, yells, etc. Also, we would like to introduce sounds that were likely to happen in a domestic environment within a **tense** atmosphere as children crying, dog barking or glass breaking. However, we also wanted to take into account the possibility of including other cases not usually consider violent a priori. For example, the sound of keys jangling or the noise produces by a shaver machine. These situations may be too particular and just would be present in few uses, but this is how we understand the problem. So, our first intention was to apply some audio processing techniques to extract low-level features, as the ones previously explained in 2.1.2. Even

though there are plenty of previous works and a lot of tools to work in this way, it was not sure which path should we had to take in order to decide what features better fitted our task. Apart from this, since the database had such a big volume it would have supposed an enormous cost of time to compute features every time we wanted to try new type of categories. Moving on, by following the advances on finding new level features already mentioned in 2.1.2, we decided to investigate new methods of extraction based on the use of Neural Networks models. Nevertheless, even thought the features obtained in this case had been more appropriated, the time consumption of training a big model was one of the aspects that did not totally convince us.

The previous selection of Audio Set as our dataset allowed us to get to know the VGGish model proposed by Google researchers for feature extraction. This system loads the parameters already learned from training with another huge dataset as YouTube-8M. This is possible due to apply transfer learning idea, explained above in subsection 2.1.3, that consists of leveraging features or weights extracted from certain models and use them in simpler ways for different tasks [25], so all the computational cost and training time is not a problem anymore. Finally, we decided to put in practice this pre trained embedding extractor by loading the given parameters so as to obtain our final input representations.

3.3. Our approach

In order to start describing our approach, we will first explain in 3.3.1 how we obtained the input data to work with by using the resources previously explained in 3.1 and 3.2, and then we will move to the implementation of the whole model in section **??**.

3.3.1. Input data

Different phases took place when trying to obtain all the necessary data from the YouTube videos specified in the database. We will explain them from the first step of deciding which classes better fit our problem to the last part in which the desired embeddings are achieved.

Violent classes

In subsection 2.2.2 it is mentioned our idea about giving the victim the right of defining her own perception of violence, so the final machine can adapt to her situation in a better way. To do so, we have taken advantage of the ontology provided by the Audio Set creators that is properly explained in subsection 3.1.2.

Our little system has been implemented based on the idea of using the parent-children relationship among the different nodes. It must go through all the branches so as to offer the victim the possibility of choosing any of the audio event categories. However, instead

of consider each label individually, this starts the way from the parent classes down to the children ones.

Let's say we begin from the class "Human sounds" that is the top level of all sounds emitted by humans contained in the dataset. The system will ask the user if she wants to advance in that direction, i.e., to go across the branches that belong to that part of the **tree**. If the answer is positive, it would go for the next class, that in this case it would be "Human Speech". It will advance this way until there were no children nodes in the actual class. When this happens, the user will be asked for adding the label to the record of *violent classes*, that will be saved in a text file so they can be read by other parts of the model further on. If the user does not want to go deeper, she will be asked to add the current class to the record. If the answer is "No", then they system will jump to the next sibling category. The corresponding flowchart is shown in figure 3.3.

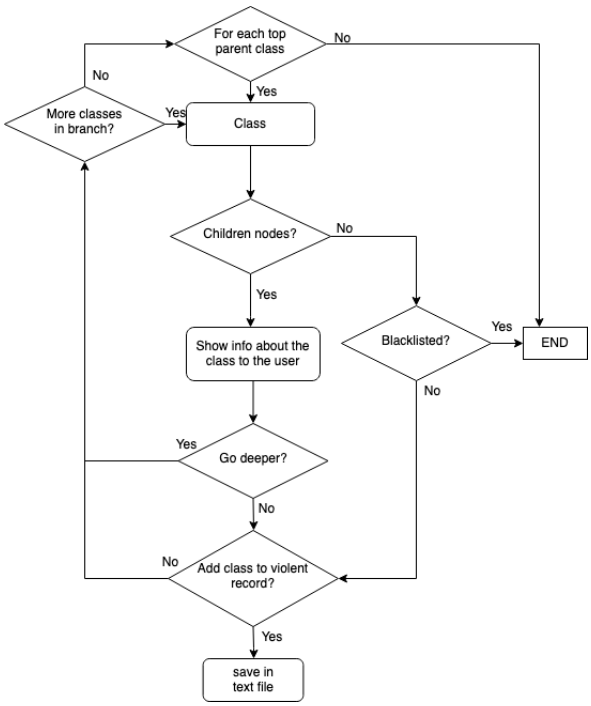


Fig. 3.3. Flowchart about selecting violent classes

This way of flowing through the different classes allows to skip those parts that does not really relate to our problem. For example, as we can see in figure 3.1, one of the top parent labels is "Natural sounds", that relates to sounds from weather phenomena. In most of the cases these classes will not be selected so the whole branch can be skipped.

Downloading videos

The following step consists in achieving audio files that belong to the chosen labels. For this purpose, we have made use of the .csv files that were explained in subsection 3.1.4. For each included video, we took its ID and build the corresponding YouTube URL. Once downloaded, we trimmed the file considering the onset and offset and, finally, converted to audio format (.wav). In our script, we can pass as a parameter the identifier of the desired classes in comma-separated format and either the number of videos per class for a balanced set or a total number of downloads for an unbalanced set. However, there might be some errors when obtaining all the data. The two most common cases are due to lack of enough videos of the desired type in the dataset or because the video is not available anymore on YouTube. When this happens, a message will be shown to the user.

It may also need to be mentioned that throughout the developing of this downloading task, a script has been coded to achieve the whole dataset in both formats, video and audio, for future works. We are not going to specify anything else since it was not finally

used.

3.3.2. Extracting embeddings

At this point, all the desired data has been already achieved to extract the embedded features that will be used to train the model. For this part, we have used the VGGish network explained in subsection 3.2.2. Sine the audio files duration is usually 10 s, and the embedding extractor gives as a result a vector of size 1×128 for each second, we will obtain a 10×128 feature matrix composed by values within the range 0 - 255. Therefore, our input feature matrix will have a size of $(numberofaudios) \times 10 \times 128$. The corresponding labels will be stored in a vector of size $(numberofsamples)$.

There are some points about the data obtained in this step that should be commented. One of them is about how long the audio files are. As it was previously indicated in 3.1.3, most of them lasts 10 s because the creators decided to set this duration for the audio events, but this can change if the video is originally shorter. For these situations, since the model is configured to have an input of 10×128, the embedding matrix of the shorter clip must be fulfilled with zero-rows to achieve the required dimensions. Even though this is not very common, there might be some silent segments that will be labelled with the category of the rest of the audio.

The other case is related to what was explained in 3.1.4. For the recently explanation of how extracting the input features from the audio files we have utilized the first manner of accessing the data, i.e., by reading the text files with the videos information before downloading. There is this second option of using the already extracted embedding features. However, these do not look exactly the same when comparing them to the ones obtained from our own extractor. This difference is due to the implementation of the given code differs from their internal production system in computing issues such as underlying libraries in the installation of VGGish and hardware **equipment**. In spite of this, the result in classification tasks are expected to be equivalent. In order to prove this assumption, we decided to try a small system with both kinds of data.

Reference from cases in which the matrix is 2D

Include json format of tfrecord files

3.3.3. Exploring differences between two types of data access

In order to check what is explained above we have decided to run a little experiment in which a small classification is performed. Also, we wanted to visualize the different features to check if we could appreciate patrons in common by using the t-distributed Stochastic Neighbor Embedding (t-SNE) algorithm, which will be explained later in this section.

Our first step consisted in determined our subset extracted from the original dataset. We thought about choosing for this small application a subset composed by three classes that could be considered violent and other tree that were non-violent. Apart from this, we paid attention to the number of samples per category to pick some class over others

Finally, we ended up picking up the labels detailed in table 3.3 and a number of 80 samples for each of them, which led us to a total of 480.

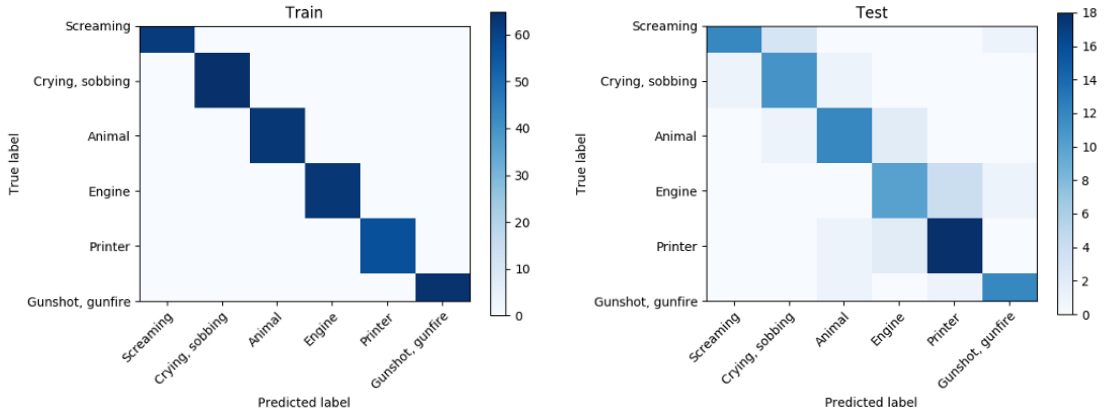
Class	**Description**
Screaming	A sharp, high-pitched human vocalization; often an instinctive action indicating fear, pain, surprise, joy, anger, etc. Verbal content is absent or overwhelmed, unlike Shout and Yell.
Crying, sobbing	Sound associated with the shedding of tears in response to an emotional state, arising from slow but erratic inhalation, occasional instances of breath holding and muscular tremor.
Gunshot, gunfire	The sound of the discharge of a firearm, or multiple such discharges.
Animal	All sound produced by the bodies and actions of non-human animals.
Engine	The sound of a machine designed to produce mechanical energy. Combustion engines burn a fuel to create heat, which then creates a force. Electric motors convert electrical energy into mechanical motion. Other classes of engines include pneumatic motors and clockwork motors.
Printer	Sounds of a computer peripheral which makes a persistent human readable representation of graphics or text on paper or similar physical media.

Table 3.3. CHOSEN CLASSES FOR A SMALL CLASSIFICATION. *SCREAMING*, *CRYING, SOBBING* AND *GUNSHOT, GUNFIRE* ARE CONSIDERED AS THE VIOLENT ONES.

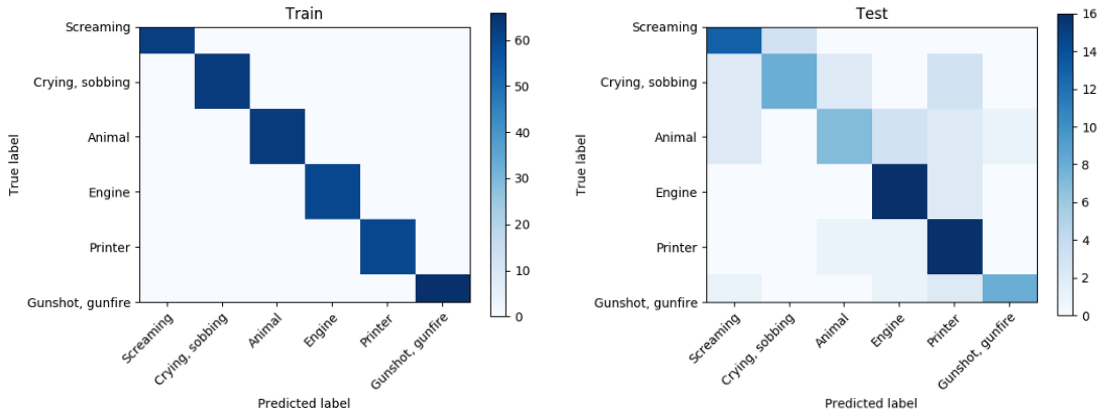
For the classification task, we decided to create a small CNN composed by few layers. Since our input data are matrices of shape 10×128, these were treated as images so the model was built with layers that perform spatial convolution [57]. The architecture is detailed in figure 3.5. We used a small kernel size of 3×3, zero padding so as not to the shape of the output and a activation of function ReLU. Two dense layers are added at the end, first one with also ReLU as activation function and the second one with *softmax* to perform the classification, and as many filters as the number of classes.

In order to measure the results, since our subset is balanced, we could have evaluated our model by computing the accuracy and the confusion matrix. More information about these metrics can be found in the appendix 3.5. In figure 3.4, the four confusion matrices corresponding to the training and test phases for both types of data are shown. Also, in table 3.4 are included the accuracy for each case. The results are more accurate when training with the embeddings extracted directly from the audio files we downloaded. When

obtaining them from the TensorFlow files, they value of the metrics indicate a worse performance. However, we opted for this second manner because the not really big difference is worth due to the mush less computational cost and time loss.



(a) Confusion matrices when embeddings are extracted from audio files



(b) Confusion matrices when embeddings are taken from *.tfrecord* files

Fig. 3.4. Confusion matrices

format / subset	**Train**	**Test**
Audio file	0.98	0.23
.tfrecord	1.0	0.72

Table 3.4. ACCURACY VALUES FOR AUDIO AND *.TFRECORD* FILES

For the training phases, it can be appreciated that there is clearly an overfitting since the accuracy is perfect. This means that the NN stop improving its capacity of learning how to solve the problem in a certain moment of the training task. Instead, it does learn some behaviour pattern that the training data follows. This impacts negatively in the model since the new data that the system will have to learn from will look different and will not follow these same rules [58]. In spite of this result, we

did not give it so much importance since we just wanted to prove with this experiment the likeness between the two types of data which can be appreciated due to the similarity of both metrics results.

t-distributed Stochastic Neighbor Embedding

Apart from the classification exercise, we wanted to see if by plotting the data samples we were really able to identify or appreciate some common patterns. In our problem, each of our samples is characterized by a matrix of features with 128 columns, which means that were are working with data belonging to high-dimensional space. Visualizing this type of data has always been a case of study for many different fields. Plenty methods have been published so as to find a solution for this task. Some of the most accepted methods consists on reducing the dimensionality of the data so this can be transformed from the high-dimensional space to a lower one and can be visualize in a common scatter plot of 2D or 3D. In particular, these techniques rely on the idea that a multivariate sample denoted as $x_i = [x_{i1}, ..., x_{in}]^T$ and considered to be a point that corresponds to a $n-dimensionality$ space, a $d-dimensionality$ space can be found, so that $d < n$, in which the data **is included**. If this is possible, then the observations can be transformed to this lower dimensional space d without any loses [59].

One of the most common and antique reduction methods is the one known as Principal Components Analysis (PCA). This follows the idea previously explained. It specifically wants to extract the *important* information of the original data by and transform them in a set formed by orthogonal variables which are actually known as principal components. This is done by multiplying the matrix data X by a projection matrix Q that contains the coefficients of the linear combinations that let perform the conversion. The projections must be orthogonal from each other and they represent the data maximum variance in descending order, being the first component the one with largest variance [60]. In fact, each of the projections correspond to an eigenvector in descending order following the value of the eigenvalue. So, the first component will be the eigenvector with the highest eigenvalue. It has been proved to be one of the most reduction dimensionality techniques nowadays. Its use is completely accepted and it is implemented in many famous software libraries. However, it presents some limitations. One of them consists on just considering linear combinations of the original data. When the relation is non-linear, a dimensionality reduction with this

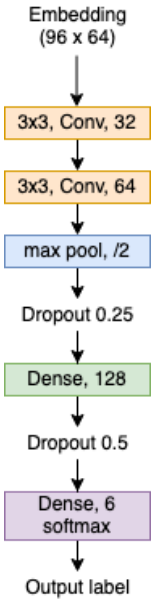


Fig. 3.5. Architecture to see how the different embeddings work

technique may result in a loss of information [61].

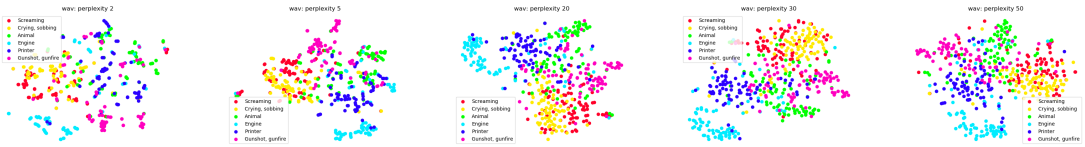
When the relation between the different subspaces cannot defined as linear, there have been developed other methods with such as t-distributed Stochastic Neighbor Embedding (t-SNE). This algorithm appears as an extension of the previously developed Stochastic Neighbor Embedding (SNE) [62]. Both are based on the same idea of a new way of measuring the similarity between samples. Instead of comparing two observations, let's call them x_i and y_j, by computing the euclidean between them, this is done by calculating the conditional probability $p_{j|i}$ of x_j being picked as a neighbour of x_i considering that the samples belong to a Gaussian distribution centered at x_i.Its depends on how far the samples are from each other, i.e., it is high when they are close and minimum when there are totally separated [63]. Apart from this, two analogous observations are created in the subspace of lower-dimensionality, y_i and y_j, and conditional probability $q_{j|i}$ is computed in this situation. It is important that, for y_i and y_j to be faithful representations of x_i and x_j, both conditional probabilities must be equal.

In order to calculate the probabilities, an crucial factor is the variance of the Gaussian distribution. There is no one unique value for this parameter, so t-SNE performs a binary search so as to find the optimal one [61]. This is also influenced by what is called the perplexity. This can be defined as an assumption of the number of adjacent neighbours for each point. It is a value that is fixed by the user, but it usually is comprehend in a range from 5 to 50 [63].

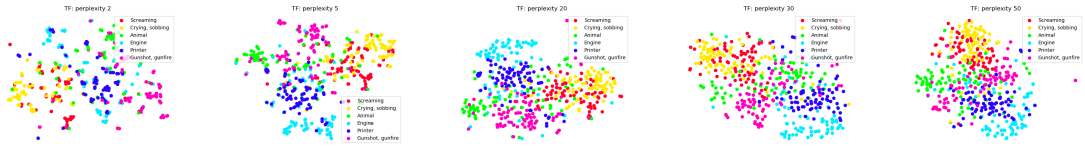
There are several considerations that should be known before looking at a representation of data from this algorithm [64]. Actually, it is not an easy task to understand this kind of plots, since the distance between points in the new subspace are not related to the real euclidean distance, which has been denoted as "The Crowding Problem" [63]. This means that the groups cannot be interpreted as real collections of data in the the original dimension. However, in order not to misunderstand the data distribution, a couple of visualizations varying the parameters usually tend to be done so the conclusions can be based on more than one result. In figure 3.6 is shown ten t-SNE outputs, five for each type of data for diverse values of perplexity from 2 to 50.

The selection of the perplexity value depends on the number of observations per class [64]. Since our subset has 80 samples for each category, we can consider that a proper value is 20 or 30. As we can see, a frontier cannot be extracted among the different labels, but we can see some grouping patterns in the data that correspond to the original labelling and helps us to confirm the similarity for the two different given types. It is true that this method should never use as an algorithm of clustering itself, but it is a good resource as a backing strategy to other results, as in this case.

3.4. Models



(a) From audio files



(b) From *.tfrecord* files

Fig. 3.6. t-SNE results from both formats with a legend that shows the labels of the data in the original 128D space

ACRONYMS

.csv Comma-separated values. 15, 16, 22

.wav Waveform Audio File Format. 22

AED Acoustic Event Detection. 12

AED/C Acoustic Event Detection and Classification. vii, 3

ASC Acoustic Scene Classification. vii, 3, 4, 6, 9

BAA Broad Agency Announcement. 8

CASA Computational Acoustic Scenes Analysis. 3

CHIL Computers in the Human Interaction Loop. 12

CNN Convolutional Neural Networks. 16, 18, 20, 24

ConvNet Convolutional Network. 16, 17

CV Computer Vision. 1, 3, 16

DARPA Defense Advanced Research Projects Agency. 8

DNN Deep Neural Networks. 7, 9

F0 fundamental frequency. 4

FC Fully-Connected. 17, 18

GMM Gaussian Mixture Models. 6

HMM Hidden Markov Models. 6

ILSVRC ImageNew Large Scale Visual Recognition Challenge. 16, 17

IPTO Information Processing Technology Office. 8

IPV Intimate Partner Violence. 10

L^3 Look, Listen and Learn. 7

LLD low-level descriptors. 4

LRN Local Response Normalisation. 17

MFCC Mel-frequency Cepstrum Coefficients. 5, 6

MID Machine ID. 15

NN Neural Networks. 21, 25

PCA Principal Components Analysis. 20, 26

ReLU Rectified Linear Unit. 17, 24

RGB red, green and blue. 17

SF spectral flux. 4

SNE Stochastic Neighbor Embedding. 27

STFT Short-Time Fourier Transform. 5, 19

SVM Support Vector Machine. 6

t-SNE t-distributed Stochastic Neighbor Embedding. 23, 26, 27

TF TensorFlow. 16, 25

UC3M Universidad Carlos III de Madrid. 10

URL Uniform Resource Locator. 22

VGG Visual Geometry Group. vii, 7, 16, 17, 18, 20, 21, 23

ZCR zero crossing rate. 4, 5

BIBLIOGRAPHY

[1] D. Barchiesi, D. D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic Scene Classification: Classifying environments from the sounds they produce", *IEEE Signal Processing Magazine*, 2015. DOI: 10.1109/MSP.2014.2326181.

[2] D. Dubois, C. Guastavino, and M. Raimbault, "A cognitive approach to urban soundscapes: Using verbal data to access everyday life auditory categories", *Acta Acustica united with Acustica*, 2006.

[3] D. Wang and G. J. Brown, *Computational auditory scene analysis: Principles, algorithms, and applications*. 2006. DOI: 10.1109/9780470043387.

[4] A. J. Eronen *et al.*, "Audio-based context recognition", in *IEEE Transactions on Audio, Speech and Language Processing*, 2006. DOI: 10.1109/TSA.2005.854103.

[5] M. Bahoura, "Pattern recognition methods applied to respiratory sounds classification into normal and wheeze classes", *Computers in Biology and Medicine*, 2009. DOI: 10.1016/j.compbiomed.2009.06.011.

[6] D. Van Nort, P. Oliveros, and J. Braasch, "Electro/acoustic improvisation and deeply listening machines", *Journal of New Music Research*, vol. 42, no. 4, pp. 303–324, 2013.

[7] A. Temko *et al.*, "Acoustic Event Detection and Classification", in *Computers in the Human Interaction Loop*, 2009, ch. Part II, 7. DOI: 10.1007/978-1-84882-054-8_7.

[8] A. Temko *et al.*, "CLEAR evaluation of acoustic event detection and classification systems", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2007. DOI: 10.1007/978-3-540-69568-4_29.

[9] E. S. Sazonov *et al.*, "Automatic detection of swallowing events by acoustical means for applications of monitoring of ingestive behavior", *IEEE Transactions on Biomedical Engineering*, 2010. DOI: 10.1109/TBME.2009.2033037.

[10] I. Potamitis, S. Ntalampiras, O. Jahn, and K. Riede, "Automatic bird sound detection in long real-field recordings: Applications and tools", *Applied Acoustics*, 2014. DOI: 10.1016/j.apacoust.2014.01.001.

[11] Y. Wang, L. Neves, and F. Metze, "Audio-based multimedia event detection using deep recurrent neural networks", in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2016. DOI: 10.1109/ICASSP.2016.7472176.

[12] T. Giannakopoulos and A. Pikrakis, *Introduction to Audio Analysis: A MATLAB Approach*. 2014. DOI: 10.1016/C2012-0-03524-7.

[13] X. Amatriain, “An Object-Oriented Metamodel for Digital Signal Processing with a focus on Audio and Music”, 2004.

[14] D. Marr, “Vision: a computational investigation into the human representation and processing of visual information.”, *Vision: a computational investigation into the human representation and processing of visual information.*, 1982. DOI: 10.1016/0022-2496(83)90030-5.

[15] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, “Detection and Classification of Acoustic Scenes and Events”, *IEEE Transactions on Multimedia*, 2015. DOI: 10.1109/TMM.2015.2428998.

[16] J. T. Geiger, B. Schuller, and G. Rigoll, “Large-scale audio feature extraction and SVM for acoustic scene classification”, in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013. DOI: 10.1109/WASPAA.2013.6701857.

[17] J.-J. Aucouturier, B. Defreville, and F. Pachet, “The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music”, *The Journal of the Acoustical Society of America*, 2007. DOI: 10.1121/1.2750160.

[18] M. Henaff, K. Jarrett, K. Kavukcuoglu, and Y. Lecun, “Unsupervised learning of sparse features for scalable audio classification”, in *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011*, 2011.

[19] J. Nam, J. Herrera, M. Slaney, and J. Smith, “Learning sparse feature representations for music annotation and retrieval”, in *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012*, 2012.

[20] K. Lee, Z. Hyung, and J. Nam, “Acoustic scene classification using sparse feature learning and event-based pooling”, in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013. DOI: 10.1109/WASPAA.2013.6701893.

[21] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, “Acoustic event detection in real life recordings”, in *European Signal Processing Conference*, 2010.

[22] J. Cramer, H. H. Wu, J. Salamon, and J. P. Bello, “Look, Listen, and Learn More: Design Choices for Deep Audio Embeddings”, in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2019. DOI: 10.1109/ICASSP.2019.8682475.

[23] Y. Aytar, C. Vondrick, and A. Torralba, “SoundNet: Learning sound representations from unlabeled video”, in *Advances in Neural Information Processing Systems*, 2016. arXiv: 1610.09001.

[24] S. J. Pan and Q. Yang, *A survey on transfer learning*, 2010. DOI: 10.1109/TKDE.2009.191.

[25] D. Sarkar, “A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning”, 2018.

[26] S. Ruder, “Transfer Learning - Machine Learning’s Next Frontier”, 2017.

[27] E. G. Krug, J. A. Mercy, L. L. Dahlberg, and A. B. Zwi, “The world report on violence and health”, *Lancet*, 2002. DOI: 10.1016/S0140-6736(02)11133-0.

[28] C. H. Demarty *et al.*, “The MediaEval 2013 affect task: Violent Scenes Detection”, in *CEUR Workshop Proceedings*, 2013.

[29] T. Giannakopoulos, D. Kosmopoulos, A. Aristidou, and S. Theodoridis, “Violence content classification using audio features”, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006. DOI: 10.1007/11752912_55.

[30] T. Giannakopoulos, A. Makris, D. Kosmopoulos, S. Perantonis, and S. Theodoridis, “Audio-visual fusion for detecting violent scenes in videos”, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010. DOI: 10.1007/978-3-642-12842-4_13.

[31] A. Ali and N. Senan, “Violence video classification performance using deep neural networks”, *Advances in Intelligent Systems and Computing*, vol. 700, pp. 225–233, 2018. DOI: 10.1007/978-3-319-72550-5_22.

[32] T. W. Chua, K. Leman, and F. Gao, “Hierarchical audio-visual surveillance for passenger elevators”, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014. DOI: 10.1007/978-3-319-04117-9_5.

[33] J. García-Gómez, M. Bautista-Durán, R. Gil-Pita, I. Mohino-Herranz, and M. Rosa-Zurera, “Violence detection in real environments for smart cities”, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016. DOI: 10.1007/978-3-319-48799-1_52.

[34] M. Bautista-Duran *et al.*, “Acoustic detection of violence in real and fictional environments”, in *ICPRAM 2017 - Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods*, 2017. DOI: 10.5220/0006195004560462.

[35] United Nations, *No Title*, 1989.

[36] WHO. Department of Reproductive Health Research. London School of Hygiene and Tropical Medicine. South African Medical Research Council., *WHO | Global and regional estimates of violence against women*. 2013.

[37] European Union Agency for Fundamental Rights, *Violence against women : An EU-wide survey*. 2014. DOI: 10.2811/62230.

[38] L. Heise, M. Ellsberg, and M. Gottemoeller, *Ending violence against women.* 1999. DOI: 10.4324/9780429269516-5.

[39] K. Beyer, A. B. Wallis, and L. K. Hamberger, "Neighborhood Environment and Intimate Partner Violence: A Systematic Review", *Trauma, Violence, and Abuse*, 2015. DOI: 10.1177/1524838013515758.

[40] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation", in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2017. DOI: 10.1109/WASPAA.2017.8170052.

[41] V. Mapell, *UPC-TALP database of isolated meeting-room acoustic events*, 2012. [Online]. Available: http://catalog.elra.info/en-us/repository/browse/ELRA-S0268/.

[42] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, "Reliable detection of audio events in highly noisy environments", *Pattern Recognition Letters*, 2015. DOI: 10.1016/j.patrec.2015.06.026.

[43] E. Fagerlund and A. Hiltunen, *TUT Rare sound events*, 2017.

[44] D. Stowell and E. Benetos, *IEEE AASP Challenge: Detection and Classification of Acoustic Scenes and Events*, 2013.

[45] E. Cakir and T. Heittola, *TUT-SED Synthetic*, 2016.

[46] C. H. Demarty, C. Penet, M. Soleymani, and G. Gravier, "VSD, a public dataset for the detection of violent scenes in movies: design, annotation, analysis and evaluation", *Multimedia Tools and Applications*, 2015. DOI: 10.1007/s11042-014-1984-4.

[47] J. F. Gemmeke *et al.*, "Audio Set: An ontology and human-labeled dataset for audio events", in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2017. DOI: 10.1109/ICASSP.2017.7952261.

[48] E. Fonseca *et al.*, "Freesound datasets: A platform for the creation of open audio datasets", in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*, 2017.

[49] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora", 1992. DOI: 10.3115/992133.992154.

[50] A. Singhal, *Introducing the Knowledge Graph: things, not strings*, 2012.

[51] Sound Understanding group, *AudioSet*, 2017.

[52] S. Hershey *et al.*, "CNN architectures for large-scale audio classification", in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2017. DOI: 10.1109/ICASSP.2017.7952132. arXiv: 1609.09430.

[53] Video Understanding Group, *YouTube-8M*, 2017.

[54] GoogleResearch, "TensorFlow: Large-scale machine learning on heterogeneous systems", *Google Research*, 2015. arXiv: arXiv:1603.04467v2.

[55] ImageNet, *Results for ILSVRC2014*, 2014.

[56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015. arXiv: 1409.1556.

[57] M. Levoy, K. Dektar, and A. Adams, *Spatial Convolution*, 2012. [Online]. Available: https://graphics.stanford.edu/courses/cs178/applets/convolution.html.

[58] H. K. Jabbar and R. Z. Khan, "Methods to Avoid Over-Fitting and Under-Fitting in Supervised Machine Learning (Comparative Study)", 2015. DOI: 10.3850/978-981-09-5247-1_017.

[59] S. Kaski and J. Peltonen, "Dimensionality reduction for data visualization", *IEEE Signal Processing Magazine*, 2011. DOI: 10.1109/MSP.2010.940003.

[60] H. Abdi and L. J. Williams, *Principal component analysis*, 2010. DOI: 10.1002/wics.101.

[61] J. Amat Rodrigo, *Análisis de Componentes Principales (Principal Component Analysis, PCA) y t-SNE*, 2017.

[62] G. Hinton and S. Roweis, "Stochastic neighbor embedding", in *Advances in Neural Information Processing Systems*, 2003.

[63] L. Van Der Maaten and G. Hinton, "Visualizing data using t-SNE", *Journal of Machine Learning Research*, 2008.

[64] M. Wattenberg, F. Viegas, and I. Johnson, "How to Use t-SNE Effectively", *Distill*, 2016. DOI: 10.23915/distill.00002. [Online]. Available: http://distill.pub/2016/misread-tsne.

[65] Scikit-learn, *Metrics and scoring: quantifying the quality of predictions*. [Online]. Available: https://scikit-learn.org/stable/modules/model%7B%5C_%7Devaluation.html.

[66] A. Mishra, *Metrics to Evaluate your Machine Learning Algorithm*, 2018. [Online]. Available: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234.

[67] F. Krüger, "Activity, Context, and Plan Recognition with Computational Causal Behaviour Models", *ResearchGate*, 2018.

APPENDIX X

3.5. Metrics

A fundamental part of a machine learning project consists of checking the performance. There are plenty of metrics to carry out this evaluation and the results will look in one way or another depending on the method utilized. The following two are the most used in this project

3.5.1. Classification Accuracy

This is a technique commonly used and it is usually referred to as just accuracy. It can be defined as the relation between the amount of right predictions and the total number on input instances [65].

$$acc = \frac{Number\ of\ incorrect\ predictions}{Number\ of\ total\ input\ instances}$$

This metric best works when dealing with a balanced dataset, i.e., the same of number of samples per class. If the problem is addressed with unbalanced data, then the accuracy value could be a higher value due to predict all the instances belong to the major class. For example, if 90% of the data are part of the same class A and all the predictions results are this class, then the accuracy value will be 90%, which apparently is a satisfying output, even though we are misclassifying all the samples from class B [66].

3.5.2. Confusion matrix

As it own name describes, the output of this type of metric consists of a matrix which shows a complete evaluation of the model. By definition, an entry i, j of the matrix denotes the amount of observations that belong to group i but are predicted as group j [65]. For example, considering a binary classification problem in which there are two classes, YES

n = 165	Predicted: NO	Predicted: YES
Label: NO	50	10
Label: YES	5	100

Fig. 3.7. Example of confusion matrix

and NO, for a test set composed by 165 samples, the matrix included in figure 3.7 is obtained.

There are four groups that can be extracted from this matrix: True positives, the samples that are predicted as YES and that is in fact their true label, True Negatives, those cases that were predicted as NO and they are originally labelled as NO, False Positives, in which the predicted label is YES but they are actually negative, and False Negatives, those in which the predicted label is NO when their original label is YES.

This metric an the one explained before, accuracy, can be related by taking the diagonal of the matrix and computing the next operation:

$$acc = \frac{TruePositives + \ FalseNegatives}{Total\ number\ of\ samples} = \frac{100 + \ 50}{165} = 0.91$$

When the classification task consists on more than two classes, a multiclass problem, a similar definition of the confusion matrix can be extended from the binary problem. Considering a certain observation C_k, the True positive part of the matrix is placed in the exact point where the column and the row of this certain observation are crossed, i.e, when the predicted label is equal to the true label. The False positives samples are placed along the column C_k for all the rows $C_0, ..., C_{k-1}, Ck + 1, ..., C_n$ which refers to all the samples that have been misclassified with the class C_k. The False negatives are, however, all the samples that originally are labelled with C_k tag but have been wrongly categorized with $C_0, ..., C_{k-1}, Ck + 1, ..., C_n$ classes. Finally, the True negative samples are distributed across all the other positions in the matrix. In figure 3.8, a good example for this explanation is shown.

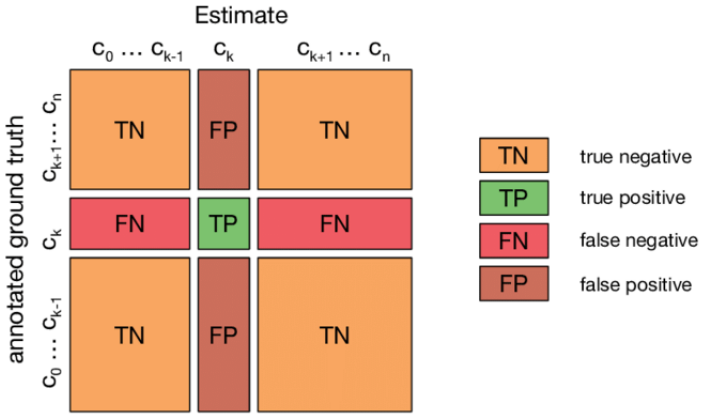


Fig. 3.8. Confusion matrix for a multiclass classification [67]