ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2017 ENST 0069

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité Sécurité numérique

présentée et soutenue publiquement par

Daniele Battaglino

le 13 Decèmbre 2017

La classification des scènes acoustiques: contributions à la recherche fondamentale et appliquée

Directeur de thèse: **Nicholas EVANS**
Co-encadrement de la thèse: **Ludovick LEPAULOUX**

Jury

M. Tuomas VIRTANEN, TUT, Tampere - FINLAND		Rapporteur
M. Emmanuel VINCENT, Inria, Nancy – France		Rapporteur
Mme/M. Bernard MERIALDO, EURECOM, Biot – France		Examinateur
Mme Christelle YEMDJI, Renault Software Labs, Sophia Antipolis – France		Examinatrice

TELECOM ParisTech
école de l'Institut Télécom - membre de ParisTech

THÈSE

ÉCOLE DOCTORALE EDITE
DOCTORAL THESIS

ACOUSTIC SCENE CLASSIFICATION: CONTRIBUTIONS TO FUNDAMENTAL AND APPLIED RESEARCH

Author:
Daniele Battaglino

Supervisor EURECOM:
Prof. Nicholas Evans

Supervisor NXP semiconductors:
Dr. Ludovick Lepauloux

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy from
EURECOM – Telecom ParisTech

13 December 2017

ABSTRACT

Acoustic context information may be used by microphone-equipped devices in order to adapt their behaviour or configuration according to a particular scenario. Recognition of such scenarios according to the acoustic context is the goal of acoustic scene classification (ASC). The choice of audio sensors, instead of alternatives (e.g. motion or light sensors), is a natural one; almost all mobile and smart devices are equipped with at least one microphone.

Almost all previous solutions to ASC rely on feature extraction approaches designed specifically for speech and music genre recognition and are thus not necessarily optimal for ASC. Further limitations of existing solutions relate to the requirements for real-time and low footprint implementations. These requirements must be met in order that ASC algorithms can be developed for low power, always listening devices.

The work reported in this thesis aims to address these limitations and hence to reduce the gap between academic and industrial research in terms of methods, protocols and metrics. Accordingly, this thesis presents the ASC problem from a dual perspective. This includes contributions in both *fundamental* research, which report contributions with respect to standard protocols and methods in addition to *applied* research, which describes contributions to the adaptation of current methods to 'real-world' applications.

The main contributions of the work include: (i) the design of ASC-tailored features which exploit spectro-temporal patterns from spectrograms using local binary pattern analysis; (ii) techniques for the automatic extraction of the most discriminative spectro-temporal patterns through the application of convolutional neural networks; (iii) the collection of a large database of realistic, low-quality audio recordings to support work in ASC; (iv) the implementation of an *always-listening*, *low-complexity* ASC system, and (v) the first investigation of ASC in an open-set scenario, a new classifier tailored to open-set classification and new protocols and metrics for the assessment of open-set ASC.

The work presented in this thesis demonstrates that greater synergy between fundamental and applied research must become the standard pathway to future work with a view to creating practical, usable ASC techniques.

RÉSUMÉ

Les informations de contexte acoustique peuvent être utilisées par des dispositifs équipés de microphones afin d'adapter leurs comportements ou leurs configurations en fonction de la scéne qui se déroule. La reconnaissance des scénarii en fonction du contexte acoustique est l'objectif de la classification des scènes acoustiques (ASC). Si il est naturel d'envisager l'utilisation des capteurs audio pour y parvenir, s'y restreindre se justifie par le fait que presque tous les appareils mobiles sont équipés d'au moins un microphone; ce qui n'est pas le cas pour d'autres types de capteurs (par exemple les capteurs de mouvement ou de lumière).

La plupart des solutions ASC reposent sur des algorithmes d'extraction de descripteurs conçus spécifiquement pour la reconnaissance de la parole et de la musique, et ne sont donc pas nécessairement optimales lorsqu'ils sont appliqués au domaine de l'ASC. Par ailleurs, rares sont les approches qui prennent en considération les exigences d'une implémentation temps réel conjointement à des contraintes de faible complexité. Or, ces exigences doivent être satisfaites pour que les algorithmes ASC développés puissent être portés sur des appareils focntionnant sur batterie et toujours à l'écoute.

Le travail présenté dans cette thèse vise à combler ces lacunes et donc à réduire l'écart entre la recherche académique et industrielle en termes de méthodes, de protocoles et de mesures. En conséquence, cette thèse propose une reformulation du problème de l'ASC sous deux aspects. Du point de vue recherche fondamentale, une première partie relate des contributions sur les protocoles et les méthodes standards. Une seconde partie traite de la recherche appliquée et décrit les contributions à l'adaptation des méthodes actuelles aux applications du monde réel.

Les principales contributions de ce travail comprennent: (i) la conception de descripteurs adaptés à l'ASC et qui exploitent les modèles spectro-temporels. Ces modèles sont calculés à partir de spectrogrammes sur lesquels une analyse de motifs binaires locaux (LBPs) est appliquée; (ii) des techniques d'extraction automatique des modèles spectro-temporels les plus discriminants par l'application de réseaux de neurones convolutionnels; (iii) la collecte d'une vaste base de données d'enregistrements de scènes sonores du quotidien; (iv) la mise en oeuvre d'un système ASC toujours à l'écoute, de faible complexité, et (v) la première utilisation d'algorithme pour l'ASC dans un scénario de classification open-set, la description d'un nouveau classificateur adapté à la classification open-set et de nouveaux protocoles ainsi que de nouvelles métriques pour l'évaluation de l'ASC poru les problèmes open-set.

Le travail présenté dans cette thèse démontre qu'une plus grande synergie entre la recherche fondamentale et la recherche appliquée doit devenir la voie standard pour les travaux futurs en vue de créer des solutions en ASC pratiques et utilisables.

The inferno of the living is not something that will be; if there is one, it is what is already here, the inferno where we live every day, that we form by being together. There are two ways to escape suffering it. The first is easy for many: accept the inferno and become such a part of it that you can no longer see it. The second is risky and demands constant vigilance and apprehension: seek and learn to recognize who and what, in the midst of inferno, are not inferno, then make them endure, give them space."

— Italo Calvino, *Invisible cities*

ACKNOWLEDGMENTS

Firstly, I would like to express my gratitude to my supervisors, *Nick* and *Ludovick*, for the continuous support of my PhD, their precious advices and patience. Their guidance helped me through the thesis and the final writing. A special thank to my managers at NXP, *JC* and *Laurent* that believed in me and created the best conditions to the fulfilment of my PhD.

Besides my supervisors, I would like to thank the rest of my thesis committee: Prof. Virtanen, Prof. Vincent, Prof. Merialdo and Dr. Yemdji, for their insightful comments and encouragement. I would also like to thank all the colleagues at EURECOM and NXP for making me feel part of a team. In particular, my NXP team mates *Rafael, Adrien, Giacomo* that worked with me and gave me useful insights for my research. A thank to the other colleagues from NXP for their kindness: *Aurelie, Sebastien, Jean-Marc, Thomas, Guilleme, Fabrice, Jean, Alex*. I would like also to thank the guys from the audio&speech group at EURECOM: *Pramod, Pepe, Hector, Massimiliano, Giovanni, Leela*.

A special thanks goes to the friends I met in these three years: *Stefano, Alberto, Cedric, Danja, Julie, Marco*. They encouraged me to do my best and they were always present.

Words cannot express the gratitude to my family: my mother *Lucilla*, my father *Gainbeppe* and my sister *Agnese*. A special thank to *Chiara*, my girlfriend, and to her parents *Gianna and Davide*. They all supported me throughout writing this thesis and my life.

Contents

List of Figures

List of Tables

Listings

MATHEMATICAL NOTATIONS

$\boldsymbol{x}$ vector

$\boldsymbol{X}$ matrix

$\boldsymbol{X}^{-1}$ inverse of matrix

$\boldsymbol{X}^{T}$ transpose of matrix

x_i i^{th} element of single vector $\boldsymbol{x}$

$\boldsymbol{x}_n$ n^{th} vector of a set

$\boldsymbol{x}_{n,i}$ i^{th} element of vector $\boldsymbol{x}_n$

$\boldsymbol{x}_{n,i}^{(t)}$ i^{th} element of vector $\boldsymbol{x}_n$ at time t

$\hat{\boldsymbol{x}}$ estimated vector $\boldsymbol{x}$

$\boldsymbol{x}^{'}$ normalised or scaled vector $\boldsymbol{x}$

x[n] discrete signal

X[k] Fourier transform of x[n]

$\Pr(\boldsymbol{x})$ probability of random variable x

$\Pr(\boldsymbol{x}, \boldsymbol{h})$ marginal probability

$\Pr(\boldsymbol{x}|\boldsymbol{h})$ conditional probability of random variable x given another random variable h

$\Pr(\boldsymbol{x}; h)$ conditional probability of random variable x given fixed parameters h

$\mathcal{F}\{.\}$ Fourier transform

$\mathcal{F}^{-1}\{.\}$ inverse Fourier transform

$|.|$ absolute value

$||.||$ euclidean norm

$||.||_p$ p-norm

α^* optimal value

FIXED SYMBOLS

$\mathcal{X}$ set of sample vectors

N_c number of samples of class c

TP_c number of correctly predicted samples of class c

$\boldsymbol{\mu}$ mean of all samples

$\boldsymbol{\Sigma}_c$ covariance of all samples

$\boldsymbol{\mu}_c$ mean of samples of class c

$\boldsymbol{\Sigma}_c$ covariance of samples of class c

θ_c set of model parameters for all classes

θ_c set of model parameters or function for class c

$\mathcal{X}_c$ set of sample vectors of class c

$\tilde{\mathcal{X}}$ decimated set of sample vectors of class c

l loss function

J cost function

F Fisher score

D_B Bhattacharyya distance

ACRONYMS

ASA	auditory scene analysis
CASA	computational auditory scene analysis
ASC	Acoustic scene classification
VAD	voice activity detection
ASR	automatic speech recognition
NLP	natural language processing
MIR	music information retrieval
AED	audio event detection
DCASE	detection and classification of acoustic scenes and events
MAP	mean average precision

MFCC	mel frequency cepstra coeffient
SVM	support vector machine
RBF	radial basis function
GMM	Gaussian mixture model
EM	expectation-maximization
HMM	hidden Markov model
MP	matching pursuit
UBM	universal background model
KNN	k-nearest neighbours
FFT	fast Fourier transform
DCT	Discrete cosine transform
DWT	discrete wavelet transform
RQA	recurrence quantification analysis
PCA	principal component analysis
HOG	histogram of gradients
BER	band energy ratio
DTs	decision trees
CI	confidence interval
KKT	Karush-Kuhn-Tucker
SVs	support vectors
SNE	Stochastic neighbor embedding
t-SNE	t-distributed stochastic neighbor embedding
KL	Kullback-Leibler
DNN	deep neural network
RMS	root mean square
LDA	linear discriminant analysis
LBP	local binary patterns
BoF	bag of features
CNNs	convolutional neural networks
MLPs	multi layer perceptrons
RNNs	recurrent neural networks
GD	Gradient descent

NAG	Nesterov accelerated gradient
ReLU	rectifier liner unit
NMF	non-negative matrix factorization
VC	Vapnik-Chervonenkis
SVDD	support vector data description
BSVs	boundary support vectors
ROC	receiver operating characteristic
AUC	area under the curve

A Chiara e Ada,
scrigni delle piccole cose.

This work was entirely funded by NXP semiconductors within the terms of the *industrial PhD* contract – convention CIFRE n. 2014/0356.

1

INTRODUCTION

Imagine closing your eyes for a moment and listening carefully to the sounds in your immediate surroundings. You may recognise specific sounds like footsteps, air conditioning, passing cars or perhaps voices. Even in the absence of visual cues, humans can identify most of the times events and sounds with acoustic cues. These acoustic cues provide information about objects which are not within the listener's field of vision. The research presented in this thesis focuses on the recognition of a specific acoustic scene by machines.

The choice of acoustic cues to recognise the surrounding environment is driven by the omnipresence of microphone in smartphones, devices with the sphere of the internet of things, wearables and hearing aid devices. While some devices are equipped with multiple, heterogeneous sensors (examples include light sensors, gyroscopes and accelerometers), acoustic sensors are the most widely used in practise. Furthermore, there is evidence [1] that context recognition using acoustic cues gives better performance than using accelerometer measurements alone. In any case, acoustic and other cues are complementary in a fusion framework.

Acoustic scene classification (ASC) aims to categorise the environment in which a device is used. The problem of recognising acoustic scenes is particularly pertinent in the case of mobile devices given their use in multiple situations throughout the course of a typical day. Here, for instance, the ringer volume of a smart telephone might be adjusted according to whether the user is on a bus, in an office or at home.

The motivation of this work stems from the continuous demand for advanced functionality by automatically adapting the device configuration to the situation or context. Moreover, the industrial nature of this PhD has conditioned tracks and axes of research. With ASC being a recent area of study, there still exists a gap between academia and industry in terms of problems, solutions, protocols and metrics; there are clear differences between lab evaluation and performance in the field. This dichotomy accounts for the structuring of this thesis in two parts; one linked to fundamental research; the other related to applied research. The final goal is to design a robust ASC system which analyses and classifies acoustic scenes in real-time on low-power devices.

This introduction is structured as follows: a definition of ASC is presented in Sec. 1.1, together with a discussion about the relationship of ASC with other domains in Sec. 1.2; examples of practical use cases are listed in Sec. 1.3; Sec. 1.4 discusses motivations and goals of this research; Sec. 1.5 details the research contributions, peer-reviewed publications and a detailed outline of the thesis.

1.1 ACOUSTIC SCENE CLASSIFICATION

ASC is the task of classifying a global scene according to ambient sounds. A scene refers to a high-level semantic concept such as *car*, *park* or *office*. ASC is a difficult task for both humans and machines without any other cues (e. g. visual). The labelling of a scene is not always clear and is open to interpretation on taxonomy. For example, different people may

describe the same scene with different high-level semantic concepts: from one angle, some distinctions are impossible to obtain from sounds alone (i. e. some cars sound like buses when only engine noise is present); from another, quiet and noisy streets may be labelled under a more general *street* concept even though they may not share common acoustic characteristics. One of the first definitions of ASC has its origins in the psycho-acoustical studies of *soundscapes* [2]. As for visual *landscapes*, *soundscapes* are also composed of ambient *background* noises in addition to descriptive *foreground* sounds. The scene is therefore a composition of background noise and foreground sounds.

Even though many computational approaches are inspired by perceptual research, there exists a notable distinction between these studies which aim to understand the human cognitive process [3] and how a machine perceives and detects sounds. The question "*Do machines hear as we do?*" exemplifies the discrepancy between human and machine sound perception. As an example, differences in perception are introduced immediately through different microphone characteristics (directivity, sensitivity, etc.). These and other such differences may lead to a representation far from that of the human auditory system.

1.2 ASC IN THE REALM OF MACHINE LISTENING

Perceptual studies [3, 4, 5] influenced the definition of an acoustic scene, which can benefit from prior research in other related domains such as speech recognition or music genre identification. These domains are focused on a specific problem related to audio even though they share common audio processing and classification techniques. More generally, these domains are part of a broader area of research, called *machine listening*, which tries to mimic the human auditory systems with machines as a whole.

As for the human auditory system, machines replicate a hierarchical process going from audio samples to a meaningful description: the audio is represented (e. g. spectrogram), organized (e. g. source separation), detected and classified. A vast majority of current *machine listening* domains (e. g. speech recognition, music genre identification, acoustic scene classification) can be interpreted according to this scheme.

Even so, the relationship between ASC and other machine listening domains appears somewhat blurred. Inspired by original work [6], current machine listening domains can be split into simpler tasks, as illustrated in Fig. 1:

- **detection**, the segmentation of useful information within a longer sequence;
- **classification**, the association of a label with the segmented information;
- **description**, the creation of high-level semantic information from the classification (e. g. from genre classification to music recommendation systems).

Following this vision, for instance, the speech domain would be split into voice activity detection (VAD) [7] (detection) followed by automatic speech recognition (ASR) [8] (classification) and then natural language processing (NLP) [9] (description) to give sense to the resulting sequence of words. The music domain would be split into music/speech separation (detection), music information retrieval (MIR) [10, 11] (classification) which may determine the genre and on top of that music recommendation (description). The ASC task fits the same formulation: the context is segmented according to some criteria, classified and then labelled to describe, for instance, a log of the different acoustic scenes encountered during a day. Audio events [12], are detected and classified before the complex scene is described as a mixture of overlapping sounds.

Each domain shares the detection-classification-description formulation, together with methods and solutions to common problems. This helps to exploit knowledge and solution

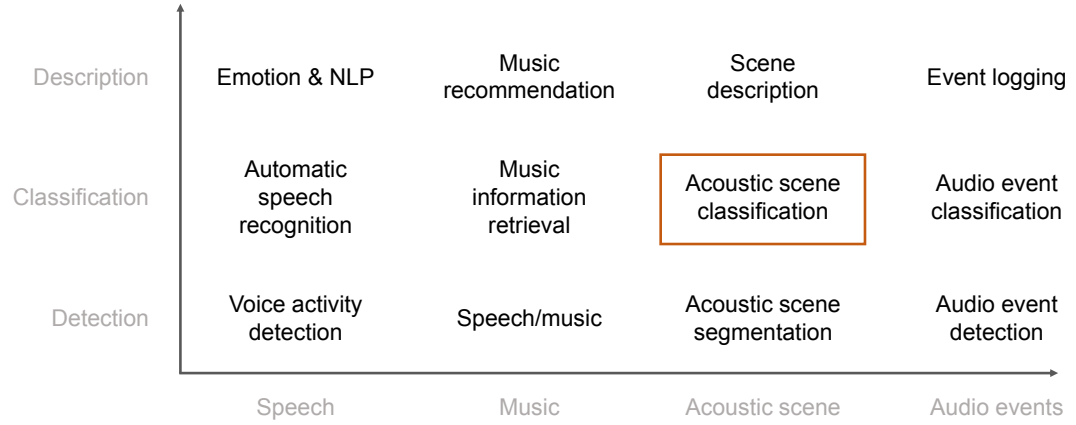


Figure 1: *Machine listening* realm is composed of different research areas varying the abstraction level of the task (i. e. detect, classify and describe). On y-axis are expressed different level of task, while on the x-axis the main research area.

from one domain in another by using, for instance, similar features or processing. As an example, ASC may exploit an audio detector algorithm to better describe a scene with audio events. At the same time, ASC may provide prior information of the scene, therefore reducing the number of possible events [12] (e. g. keyboard taping is more probable in an office rather than in a street).

ASC is a relative new topic in machine listening. The presence of consolidated machine listening domains (e. g. speech, music) has initially allowed researchers to adapt methods to ASC. At the same time, the availability of a huge number of different methods has partially limited a broader discussion on the specifics of the ASC task.

1.3 APPLICATIONS OF ASC

Applications which can directly benefit from ASC encompass existing technologies from smartphones to hearing aids:

Context-awareness devices include an *always-listening* capabilities to adapt behaviour to the surrounding situation [13]. Examples include the adaptation of a ringer volume according to whether the user is on a bus, in a office or at the cinema [14]. Evidence [15] shows that the capability to associate a behaviour to a context is particularly convenient for users. Another example of practical applications is reported in [16], where wearable devices adjust the rate (or intensity) of notifications depending on the context. The cost of being distracted by a device may be high: imagine receiving many notifications in the *car* while driving, at the *restaurant* with other people or while crossing the *street*. The decision to notify or not and how to notify the user, should be made with consideration for the current context.

Listening robots use information of "*where I am*" to switch behaviour. Especially in high mobility conditions, prior information of where the robot is located helps in defining the most appropriate actions to be performed [17]. Concrete examples may use ASC to change robot speed whether it is located indoors or outdoors [18].

Automatic data tagging exploits context similarities for automatically labelling audiovisual data. There exists a huge amount of multimedia content not segmented, neither labelled, whose manual tagging would be practically impossible. Combining video, image and acoustic scene information would allow to tag automatically a huge amount of material. This material could then be used to re-train ASC with larger datasets [19].

Hearing aids adapt their configuration to the user's environment, such as a quiet *office*, *restaurant* or *music hall*. Current hearing aid solutions are tuned according to general acoustic environments that do not adapt quickly to changes in context [20]. ASC solutions could be used to improve audio quality and to enable *context-based* configurations.

In all of the above applications, ASC is essentially a preprocessing step which provides prior information to other systems. It can inform speech recognition engines on the type of acoustic noise to improve performance [21]; it can help noise-monitoring [22] or source separation systems [23]. In addition, different applications may fuse audio cues with other sensor information such as acceleration, pressure or light [24] to obtain more accurate and confident predictions of a context.

1.4 MOTIVATIONS AND GOALS

The investigation of ASC is motivated by many factors, linked to the practical scope of this PhD: ASC research was driven by bridging the gap between fundamental and applied research. The main goal of this work is to deploy context-awareness systems which can help users in their daily lives. Considering that context-aware algorithms are to be implemented for low power devices, computational efficiency and real-time processing assume a strategical role. Dealing with channel variation or adapting metrics and evaluation protocols are other examples.

The choice of focusing on application to embedded devices rather than full power or cloud solutions is strategical in context-awareness: unreliable data connections and power implications of continually streaming audio to a remote server makes cloud solutions impractical. Moreover *always-listening* devices may impact user privacy by sending sensitive, personal information contained within audio recordings. Cloud solutions require the sharing of context information such as speech, music and other sound events which can be used to track individuals and their activities [25]. Under this assumption, ASC approaches that run locally on the device have clear advantages.

1.5 CONTRIBUTIONS

The structure of the thesis reflects the nature of the contributions regarding both fundamental and applied research. The outline is illustrated graphically by a *mind-map* in Fig. 1. Fundamental research is the focus of Part 1 (to the left of the Fig. 1) which describes the contributions between the first public challenge on ASC in 2013 [6] and the second in 2016 [26]. The sequence of the chapters follows temporally these two *milestones*, relating the the public DCASE challenges in 2013 and 2016. Applied research is the focus of Part 2 (to the right of Fig. 1) which deals with practical implications of ASC in real-world scenarios. Contributions of this part include the adaptation of ASC solutions to work in streaming fashion with reduced complexity.

The work reported in this thesis resulted in several publications:

- publication 1 (conference paper): "Acoustic context recognition for mobile devices using a reduced complexity SVM", 2015 IEEE European Signal Processing Conference (EUSIPCO);

- publication 2 (conference paper): "Acoustic context recognition using local binary pattern codebooks", 2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA);

- publication 3 (workshop paper): "Acoustic scene classification using convolutional neural networks", 2016 IEEE Detection and Classification of Acoustic Scenes and Events challenge (DCASE);
- publication 4 (conference paper): "The open-set problem in acoustic scene classification", 2016 IEEE Workshop on Acoustic Signal Enhancement (IWAENC);
- publication 5 (conference paper): "Baby cry sound detection: a comparison of hand crafted features and deep learning approach", 2017 Springer Engineering Applications of Neural Networks conference (EANN);
- publication 6 (patent): "Acoustic Context Recognition using Local Binary Pattern Method and Apparatus", US Patent App. 15/141,942
- publication 7 (patent): "Embedded car detector based on acoustic sensor", EU patent App. under approval.

Part 1 starts with Chapter 2 which describes the state of the art of ASC in 2013, at the time of the first public challenge in ASC. Together with a public challenge, a dataset was also released. Albeit being a huge step towards the standardisation of the ASC task (data, protocols, evaluation metrics), standard methods were still based on features mainly designed for speech or music (e. g. mel frequency cepstra coeffient (MFCC)). The winning system of this challenge, in fact, estimates and models recurrent patterns in MFCCs. This system and its main limitations are discussed in Chapter 3, where a first baseline is also presented. Possible ways to evaluate and visualize audio features are presented in chapter 4 which leads to the design of new features. To date, almost all existing approaches to ASC are based on traditional features designed for other domains. Even so, experiments show that these features may not be sufficiently discriminative for the ASC task.

Given the focus on ASC-tailored features, the complex acoustic structure of a scene is found to be represented by local spectro-temporal patterns, extracted directly from spectrogram (publication 2 and 6). Consequently the idea of extracting spectro-temporal patterns is then exploited using a particular topology of deep neural networks as reported in Chapter 6. This contribution (publication 3) has been submitted and publicly evaluated within the context of the DCASE 2016 evaluation whose main results and trends are presented in Chapter 7.

The outline of Part 2 is summarized as follows: Chapter 8 describes practical issues of ASC. The NXP dataset, while proprietary, is considered a contribution in the context of an industrial PhD. The data contained in this dataset can be used not only for ASC, but also for other related tasks (event detection, mixing speech with acoustic scene recording to learn more robust model, etc.). Computational constraints in terms of complexity and memory are addressed in Chapter 8 with an additional contribution including a reduced complexity ASC system (publication 1).

One of the biggest limitations of current ASC systems involves its application to closed set problems. In practice, ASC applications are open set in nature, where the number of classes during evaluation is unbounded. Contributions include the proposal for a new approach to the evaluation of ASC solutions with an open-set approach, as reported in Chapter 9. This contribution (publication 4) presents the ASC problem as an acoustic scene detection where a small number of *known* scenes are detected in a larger universe of *unknown* classes. Conclusions in the final Chapter 10 collect thoughts and findings from fundamental (Part 1) and applied (Part 2) research, and describe ideas for future research.

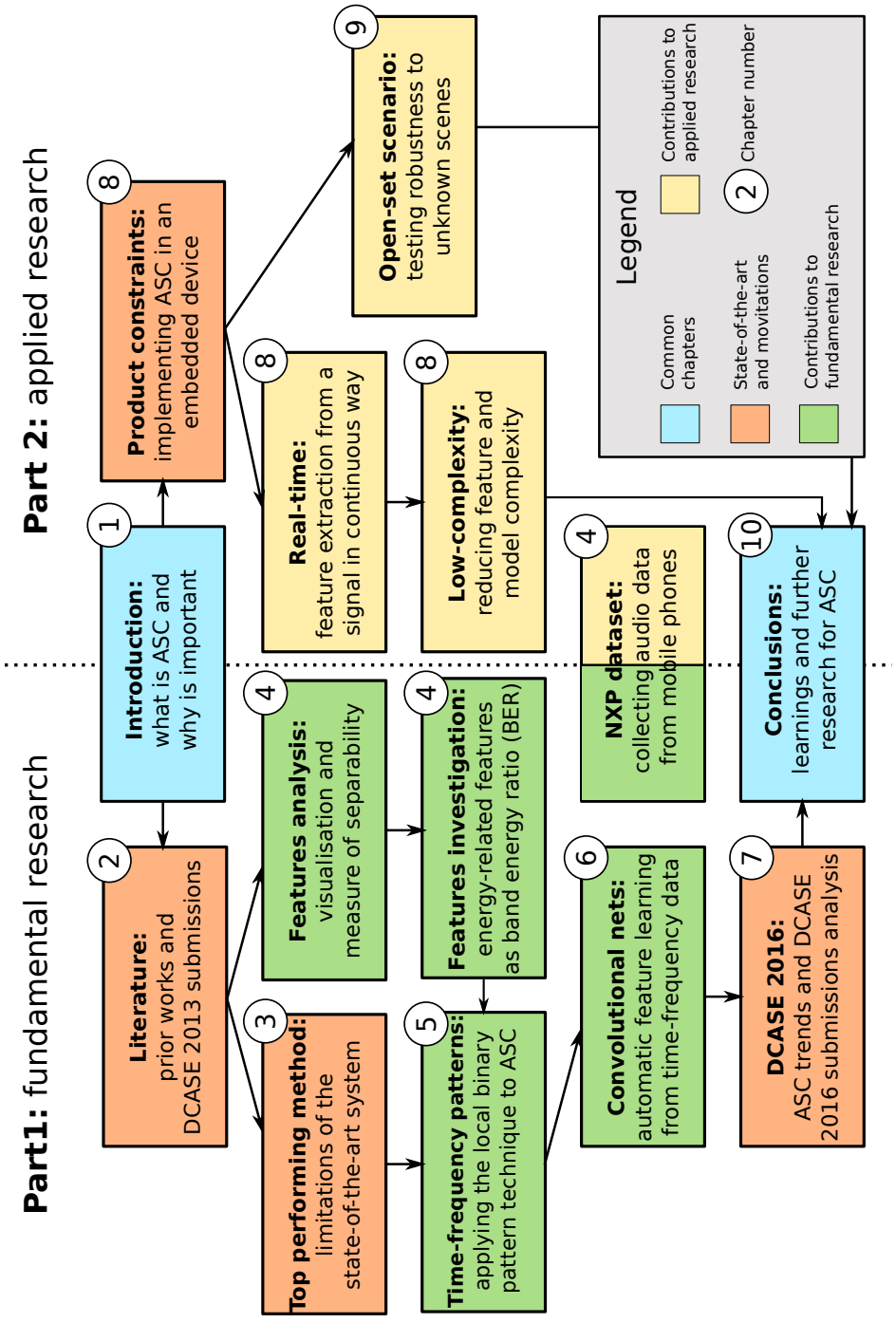


Figure 2: *Mind-map* of main blocks composing the thesis. The legend in the bottom-right helps to read the entire picture. Numbers in the upper right corner represent the chapter index.

Part I

FUNDAMENTAL RESEARCH

2

LITERATURE REVIEW

ASC covers works spanning a period of time from the first work in 1997 to the more recent in 2014, the start of this thesis. This chapter offers a view of the historical background and the most influential methods in ASC literature. As for the majority of machine learning systems, ASC solutions are composed of successive blocks. These blocks treat the audio input (preprocessing), extract a compact representation from it (feature extraction and feature post-processing), learn an inference model (classifier) and test performance on unseen samples (testing). This process is properly defined in Sec. 2.1. A timeline of ASC works between 1997 to 2013 is illustrated in Sec. 2.2. In Sec. 2.3 a detailed explanation of detection and classification of acoustic scenes and events (DCASE) 2013 database and associated challenge are reported. Works submitted to DCASE 2013 challenge are then grouped according to these blocks: features extraction in Section 2.4; feature post-processing in Sec. 2.5; classifier and testing in Sec. 2.6 and 2.7. A comparison of the ASC systems performance is then presented and discussed in Sec. 2.8.

2.1 MAIN BLOCKS OF ASC

The task of recognising and classifying an acoustic environment is generally to assign a semantic label to a certain portion of an audio signal. The labelling of a generic acoustic scene is open to interpretation: a taxonomy shared by all researchers in this domain is, therefore, difficult. Current approaches treat ASC as a *supervised classification* problem, where the taxonomy of possible categories is bounded and known in advance depending on the application (e. g. a hypothetical transport scene classifier may use a subset of categories such as *bus*, *car*, *train* and *plane*). Even though the majority of current methods uses supervised classification, alternative solutions have also been reported in an unsupervised manner [27, 28, 29], where the scenes are deduced during the processing (i. e. clustering audio samples). These unsupervised approaches require a huge amount of data to extract the underlying data structure. Therefore, the lack of a common dataset has initially blocked investigation in this direction.

In its supervised formulation, ASC does not differ from other standard machine learning problems [30] which are a concatenation of specific domain knowledge (acoustic properties of a scene) with statistical inference over the data (model training and testing). Hence ASC can be split into simpler blocks, such as those illustrated in Fig. 3, whose details are listed below:

1. **preprocessing** transforms and prepares the acoustic wave for the further processing (feature extraction and classification). Each acoustic wave is described as a variation of sound pressure across time. This sound pressure is measured by a digital microphone at a certain *sampling frequency*. The resulting digital signal is discrete in amplitude and time [31]. Examples of preprocessing operations comprise filtering, segmenting a long recording in equal-size clips or the averaging a two-channel stereo signal;

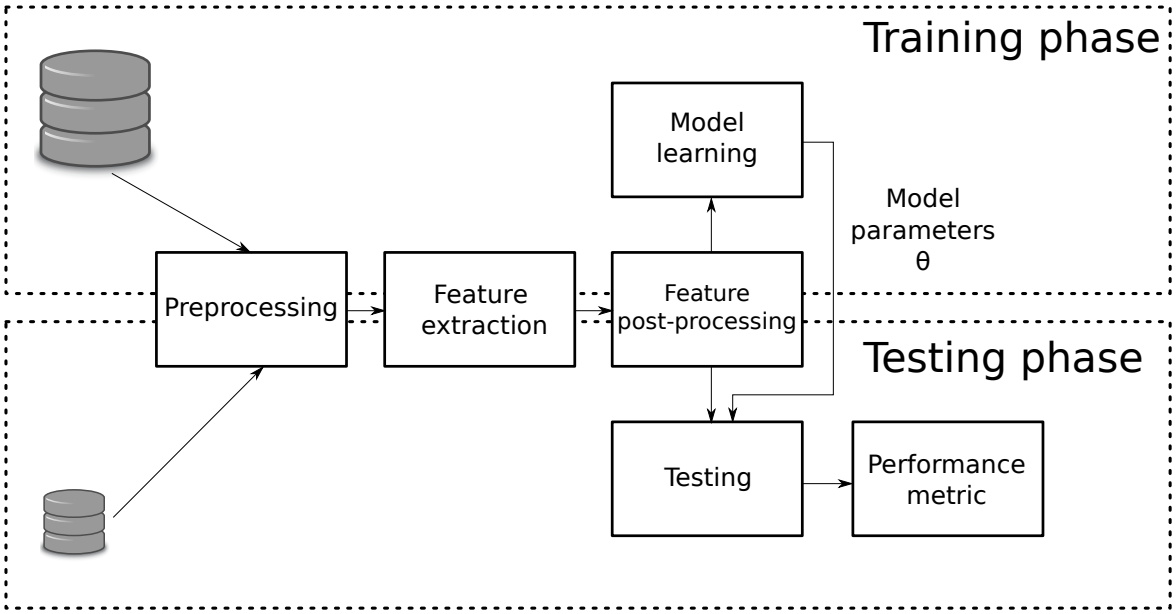


Figure 3: The main blocks composing an ASC solution: first the waveform is processed, then feature vector is extracted and processed. After that, a model is learned from all the feature vector samples and then tested on unseen data.

2. **features extraction** has the role of describing an audio wave in a more compact way by reducing the number of dimensions needed to represent it. Standard approaches consist of splitting an audio input into *frame* of 20ms over which features are calculated. Extracting features over these short-term frames ensure that each feature vector represents a statistically stationary segment of the original audio signal;

3. **features post-processing** additionally enhances a particular aspects of the original features. As an example, time derivatives of consecutive frame-based features can be added as additional information on time evolution of an acoustic scene;

4. **model learning** recognises patterns in the features space. Let $\boldsymbol{x}$ be a continuous random variable whose value is the feature vector and with θ_c the model of the c^{th} class. The goal of model learning is to "learn" this relation. As we will see in the next sections, there exist different methods to estimate this relation: some of them aim to learn the underlying distribution of the training data; others aim to maximise the separability between class samples;

5. **testing** assigns a feature vector $\boldsymbol{z}$ to the most likely class c. We define as *posterior* probability $\Pr(\theta_c|\boldsymbol{z})$ the probability of a class model θ_c given the feature vector $\boldsymbol{z}$. All previous preprocessing (feature extraction and feature post-processing) are applied to the test sample $\boldsymbol{z}$. The decision of the most likely class for $\boldsymbol{z}$ corresponds to the predicted class $\hat{c}$ which maximises $\hat{c} = \arg\max_c \Pr(\theta_c|\boldsymbol{z})$, where the model θ_c is obtained from model learning. Consequently, each sample in the test set will be assigned to one of C classes;

6. **performance metric** estimates classification accuracy and is defined as the ratio between the correctly predicted samples and the total predictions for all C classes. The confusion matrix, instead, displays directly the misclassification between classes. In the $C \times C$ confusion matrix, each element corresponding to the c^{th} row and $\hat{c}^{th}$ column represents the true class c which has been predicted as $\hat{c}$. From the resulting confusion matrix, the global accuracy is found by summing the elements on the diagonal divided by the total number of elements.

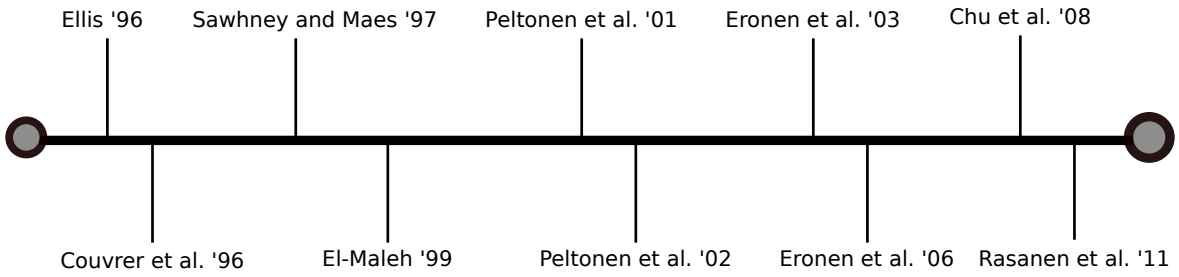


Figure 4: The main contributions in ASC before DCASE 2013.

ASC problems usually involve a huge number of possible scenes ($C > 10$). Accuracy is the standard metric because summarises the performance of a multi-class system with a single value. Nevertheless, it is heavily influenced by the balance between class samples. This is called "accuracy paradox" [32] and affects unbalanced datasets.

In order to deal with this paradox, the mean average precision (MAP) [33] metric has been preferred to standard accuracy. This variant of the metric calculates the global accuracy as a sum of single class precisions:

$$\text{MAP}\ (\%) = \frac{100}{C} \sum_{c=1}^{C} \frac{TP_c}{N_c}, \tag{1}$$

where C is the number of classes involved, TP_c stands for the number of correctly predicted samples divided by the total elements N_c of the c^{th} class. Consider an example with Class 1 ($N_c = 10, TP_c = 0$) and Class 2 ($N_c = 1000, N_c = 1000$) and calculate the MAP and accuracy metrics: $MAP = (\frac{0}{10} + \frac{1000}{1000})\frac{100}{N} = 50\%$; $accuracy = (\frac{0+1000}{1000+10})\frac{100}{N} = 99\%$. MAP reports a less biased metric, while the standard accuracy shows an unrealistic measure of performance (Class 1 has no correct predictions). MAP will be the reference metric of the system performance in the following of this thesis, because it will be perfectly comparable with the standard accuracy performance in the case of balanced datasets while being less biased in presence of unbalanced datasets.

In conclusion, the vast majority of ASC methods follows the aforementioned structure differing by the choice of preprocessing, features and classification methods. Even systems which may appear very different on the surface, still fit this common interpretation.

2.2 HISTORICAL BACKGROUND OF ASC

Contributions from the first work in 1997 until the DCASE challenge in 2013 are depicted on a timeline in Fig. 4. Several approaches have been proposed in the past to classify sounds and acoustic scenes, supported by psycho-acoustical studies [5]. One of the most relevant conclusions of these studies is that our auditory system relies on a sound-memory capable of associating sounds to a meaningful environment. In light of this, Ellis [34] in '96 proposed to describe an acoustic scene as a mixture of simpler building elements. In the same year, Couvreur et al. [34] investigated an automatic recognition of environmental noise sources (such as *car*, *truck*, *plane*) based on their global acoustic properties. This approach was further developed by El-Maleh et al. [35] in '99 using spectral features and a Gaussian classifier.

The first method specifically addressing the ASC problem relates to a technical report of Sawhney and Maes in '97 [36]. The authors recorded a small dataset composed of people voices, subway, traffic and other classes. From these recordings they extract features based

on psycho-acoustical filters, employing a recurrent neural networks classifier. They report a classification accuracy of 68% over 5 classes.

Few years later in '01, Peltonen et al. [37] were showing that humans identify a scene with typical sound events, such as a click, a door slam or a car engine. Tests performed on 19 individuals showed an overall 70% classification accuracy over 25 classes. The huge variation of accuracies between classes (it varies from 32% to 100%) depends on acoustic cues present in the scene: when the sounds in the scene are determinant in distinguishing a class from another, accuracy was higher. As expected, silent environments without prominent sounds do not bring sufficient information for the classification. This leads to the conclusion that an ASC task needs a longer excerpts of information to define its prediction where the probability of finding prominent sounds increases over time.

Influenced by these cognitive studies, Peltonen et al. [38] in '02 experimented the recognition of 6 meta-classes built over 17 starting ones. *Vehicle*, for instance, is a meta-class comprising *car*, *planes*, *bus*. The second contribution correlates the classification accuracy with the duration of a scene. As expected, a classification integrated on a longer time contains more prominent information, as previously mentioned in [37]. Therefore, an ideal length for having stable classification results suggests a 30-40 seconds of signal. In spite of these observations, the most relevant aspect of Peltonen's research was to apply for the first time MFCC and Gaussian mixture model (GMM) to the ASC problem, achieving a 68% accuracy over 17 classes. The adoption of MFCC-GMM provided a baseline system for future research. Continuing Peltonen experiments, Eronen et al. [39] in '03 exploited the temporal evolution of the acoustic scene to improve the MFCC-GMM baseline system, by using a 2-state fully connected hidden Markov model (HMM). This system was compared to human ability to recognize 18 classes and 6 meta-classes (e. g. outdoor, vehicles, indoor, etc ...). The recognition accuracy of HMM system is 61% over 18 classes against the 69% of human listening tests.

Another research axis questioned the scene taxonomy: which are the connections between everyday personal experience and collective assessment through a high-level linguistic concept? Dubois et al. [40] in '06 investigated this association between high-level concepts and acoustic scenes. The research showed that individuals classify acoustic scenes on the basis of prior experience. To enforce this perspective, a further study was conducted by Tardieu et al. [41] in '08 about the human organization of acoustic cues in increasing levels of abstraction. In the context of a *rail station* acoustic scene, they demonstrated that people use local acoustic cues (human activity) and global information (reverberation, intensity) to hierarchically construct an acoustic scene. The same idea has been recently proposed by Torija [42] in '13. By using 15 acoustic descriptors, an acoustic scene is composed by these building elements.

The definition of a suitable set of features for ASC became the subject of research for Chu et al. [43] in '08. In their work, a new way of extracting features, called matching pursuit (MP), was applied: the audio signal is decomposed by selecting the closest basis from a dictionary previously created. Then each audio signal is represented as a linear combination of these dictionary atoms.

According to Räsänen et al. [1] in '11, the use of audio classifier combined with acceleration brought to better context classification performance. Instead of fusing low-level sensory information (i. e. directly combining features coming from acoustic and acceleration sensors), only classification predictions are combined. In fact, the final prediction is a weighted-sum of single predictions coming from audio and acceleration classifiers. A similar intuition has been adopted for fusing visual and acoustic cues by Lee et al. [44] in '12.

A full hierarchical approach was proposed in Feki at al. [45] in '11. In this top-down approach, each audio streaming was classified into speech, music or environmental sounds.

If the audio streaming did not contain either speech or music was further classified according to the most probable acoustic scene. This approach decomposes a global classification problem into simpler sub-classification tasks, from high-level concepts until single sound events.

In term of reproducibility and comparability of results, ASC domain was lacking of a common dataset. Before 2013, each work mentioned above was using a different dataset (with a different number of classes and recording conditions). The first dataset on DCASE was released in 2013, associated to a public evaluation of ASC methods. Sec. 2.3 details protocols and rules of this challenge. To summarise, problems coming from this section anticipate those of the following chapters, in particular: i) the bottom-up or top-down strategy to solve an ASC task, the former initially expressed by Ellis [34] and the latter by Couvreur [22]; ii) the capacity of human listeners to distinguish different scenes (Peltonen [37], Eronen [39]); iii) the class taxonomy from Dubois [40]; iv) the temporal recurrence of acoustic scene in Eronen [46]; v) ASC-tailored features in Chu [43].

2.3 DCASE 2013

Recent trends in the signal processing community have promoted reproducibility as a fundamental aspect of scientific research. This attitude relates to sharing code, datasets and tools in order to reproduce exactly experiments described in papers: examples include music retrieval [47], speech recognition [48], source separation [49], speaker authentication [50] and anti-spoofing for speaker authentication [51].

Works prior to DCASE 2013 were typically performed with variable data (quality of the microphone, types and number of classes are some examples). As a result, most works were assessed using different databases of recordings. DCASE challenge dataset, whose main objective was to support reproducibility and comparisons with other solutions, addressed exactly this issue. The DCASE 2013 database contains recordings of the following acoustic scenes: *bus*, *busy street*, *office*, *open-air market*, *park*, *quiet street*, *restaurant*, *supermarket*, *tube* (underground train) and *tube station*. The database is split into two separate datasets of the same size, one publicly released and a second which is reserved for evaluation. Each of those datasets contains 100 recordings of 30-second audio files (WAV, 2-channel stereo, 44.1 kHz, 16-bit) with 10 samples per class. The development dataset was already provided to participants with ground truth labels identifying each scene. Training, validation and testing of the system parameters are performed on a split of the development set. The split is obtained with a 5 fold cross-validation which allows the creation of 5 non-overlapping portions of 80 recordings for training, and 20 for testing. The cross-validation covers the full datasets (meaning that each recording will be at least in one of the testing split). The result of the stratified 5-fold is illustrated in Fig. 5.

Once validated on the development set, the algorithm is submitted to be tested by the organisers on the withhold evaluation set. The evaluation protocol employs the same cross-validation used for the development set so that each of the 5 folds contains 80 training files and 20 testing files. This is done for two main reasons: first, the possibility of selecting a good subset by chance is reduced; second, the composition of recordings from each acoustic class is balanced. Before doing any other quantitative analysis of the data, it is necessary to listen to all of 100 wave files which are part of the DCASE 2013 development set. This helps to have a general overview of the data and which characteristics of the scene can be represented by the features. The following provides a qualitative description of each of the acoustic classes in the DCASE 2013 development dataset.

bus characterised by the engine noise, mainly concentrated below 300Hz. In some recordings, there are some door beeps (more than 2000Hz) which are repeated during few seconds.

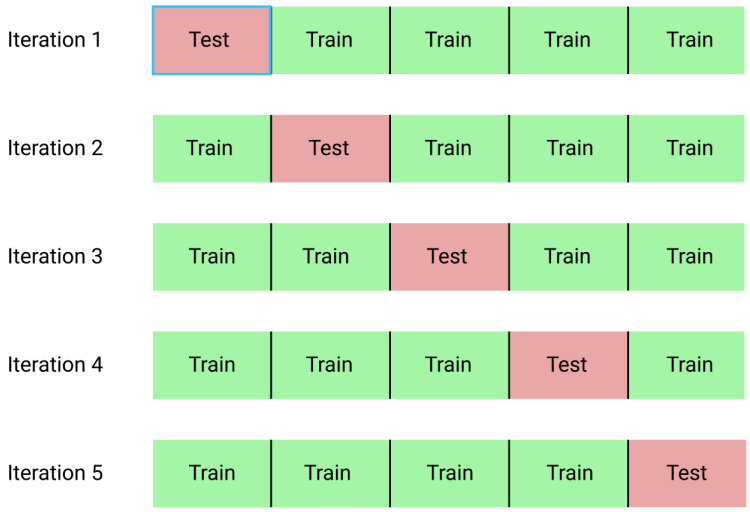


Figure 5: The 5-fold protocol in DCASE development and evaluation set. The split is always done at file level in order to obtain two completely disjoint training and testing sets.

Prominent sounds include gear changes or acceleration. There are also voices (from both passengers and artificial voices announcing the stops). Overall energy level is concentrated in lower frequencies.

busystreet similar to the *bus* scene, but energy is distributed more equally across all frequencies. There are also sounds of traffic (passing car, engine, breaks) with relatively less energy.

office low energy due to a predominance of silence. There are sparse events, such as keyboard taping, cough, whispers, mouse clicks. The only repetitive sounds in the scene are a printer or an air conditioning fan.

park Similar to *office*. This acoustic scene is characterised by quietness and silence interrupted by wind noise, steps sounds, bells and bird tweets. Sounds of nature are the most prominent, while other sounds (i. e. traffic) can be heard in the far-field.

quiet street acoustically close to *park*. The two are difficult to distinguish even for human listeners. Even so, some sounds are prominent such as walking on asphalt which produces a specific sound. For some recordings, the presence of traffic closer to the microphone suggests a street noise rather than park noise.

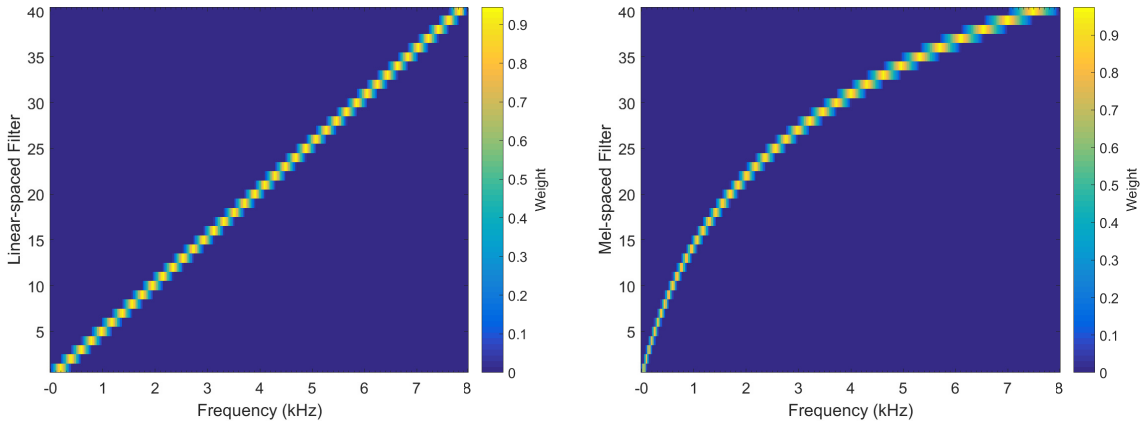
restaurant characterised by highly energetic impulsive sounds coming from forks, knives, plates. The background noise comprises a mix of overlapping voices.

supermarket similar to *restaurant* noise. Prominent sounds include the beginning of a cashier register and radio music.

tube identified by doors, cyclic sounds of the carriages, doors opening beeps and artificial or registered voices announcing the stops.

tube station similar to *tube*. The most deducible difference is a stereo effect of trains passing from one channel to another.

The DCASE 2013 database was publicly released together with a baseline ASC. This baseline is based on MFCC feature extraction and GMM classifier.



(a) (b)

Figure 6: The difference between linear (a) and Mel-scale (b) filters-bank. The Mel-scale filters-bank has more resolution in the low frequencies.

2.4 FEATURES

The general success of machine learning techniques to solve this kind of classification problems relates to the form of data. Feature extraction transforms raw data in a new space of representation where underlying structure and patterns are easier to detect. In the following subsections, DCASE methods sharing similar characteristics have been grouped under a specific features *family*. Examples of these *families* comprise low-level audio features, cepstral or spatial.

2.4.1 *MFCC: a baseline feature extraction*

The utilisation of MFCC as audio feature led to advancements in speech and speaker recognition, music genre classification among others. It has been used also in ASC as a reference feature extraction method. This section will explain the reasons for its adoption. Let $x[n]$ be the signal after being framed with a window of N samples and $|X[k]|$ the absolute value of its fast Fourier transform (FFT). Frequency bins corresponding to a certain frequency range are mapped into the Mel frequencies bands, which approximate the human pitch perception. The Mel filters-bank has a higher resolution at low frequencies than at high frequencies. The difference between a linear and a Mel filters-bank is shown in Fig.6: frequencies in $[0, 5]$kHz range are mapped on the first 26 linear-spaced bands (left inset of Fig.6); in Mel-spaced bands, 26 bands represent a $[0, 3]$kHz frequency range, resulting in a higher resolution of low frequencies (right inset of Fig.6). The magnitude coefficients of FFT are then multiplied with the corresponding Mel-filter weights and the results accumulated.

The m^{th} filter to be applied to a specific frequency bin k is identified with $H_m[k]$. M stands for the total number of Mel-scale filters and K the total number of frequency bins. Hence, the log-power at each of the Mel frequencies is calculated according to:

$$S[m] = \ln\left(\sum_{k=0}^{K-1} |X[k]|^2 H_m[k]\right) \quad 0 \leqslant m \leqslant M, \tag{2}$$

where m typically varies from 20 to 40 depending on different implementations and tasks. Discrete cosine transform (DCT) is the last step in the MFCCs calculation. It encodes the

rate of change in different spectrum bands as a sum of cosines at different frequencies and amplitudes:

$$x_i = 2 \sum_{m=0}^{M-1} S[m] \cos\left(\frac{\pi i}{2M}(2m+1)\right) \quad 0 \leqslant i \leqslant D, \tag{3}$$

where M is number of mel-scale filters, m the current m^{th} filter, D is the dimensionality of feature vector $\boldsymbol{x}$ at the i^{th} dimension. Any periodicities or repeated patterns in the Mel-log spectrum will be represented with the corresponding MFCC coefficients. Thus, one reason of the success of MFCCs for ASC stems from representing general properties of the spectrum with a relatively small number of coefficients. There exist eight different ways to express the DCT, in particular related to the period of the cosine. The DCT of type II extends a signal sequence to match a symmetric period cosine of $2M$. This is demonstrated to have a higher *energy compaction* [52]: MFCCs coefficients are concentrated at lower indices than other DCT transformations. From a machine learning point of view, DCT-II energy compaction is preferable because it gives a higher fidelity representation of the original signal with fewer coefficients.

MFCC is an approximation of a homomorphic operation [31], since MFCCs are obtained through a reverse order of summations and logarithms. It would have been if Eq. 2 had been written as $S[m] = \left(\sum_{k=0}^{K-1} \ln |X[k]|^2 H_m[k]\right)$. The advantage of performing the logarithm of the output of filtered energies $|X[k]|^2 H_m[k]$ is indeed to be more robust to noise. On the opposite, doing the logarithm within the sum would amplify the small variations produced by noise before the Mel-filtering.

MFCCs have been proven to be particularly pertinent in the speech domain, because they well approximate the separation of the glottal excitation (source) from the vocal tract (filter). This separation is obtained by selecting only the first MFCCs coefficients, because the logarithm separates the source and the filter with a simple subtraction. This operation is equivalent to take the first 13 out of M coefficients.

With MFCCs being originally designed for speech, they rise criticisms when applied to different domains such ASC. The first criticism relates to mel filter-bank resolution in low frequencies. While being beneficial for some acoustic scenes, a better resolution in low frequencies may affect other classes. The second criticism concerns the choice of selecting the first 13 coefficients. In fact, 13 MFCCs encode information about vocal tract which is essential for speech or speaker recognition. Using the same number of MFCCs to a non-speech-based task as ASC may not be optimal. The third criticism deals with robustness of MFCCs in presence of overlapping sounds. DCT represents the rate of change in different spectrum bands. By changing a value of a specific band, several DCT coefficients can change.

MFCCs for bus noise and speech signals are illustrated in Fig. 7. Difference are visible in the lower coefficients for bus noise, where the noise of the engine creates harmonics captured by the coefficients 1-2; the speech, on the other side, has a more complex structure reflected by a larger and higher number of coefficients.

MFCC has been largely employed in DCASE 2013 challenge [53, 54, 55, 56, 57, 58]. Nevertheless, some specific works extract ceptral coefficients from different time-frequency representations: in [58], for instance, discrete wavelet transform (DWT) is used as an alternative to standard FFT spectrograms.

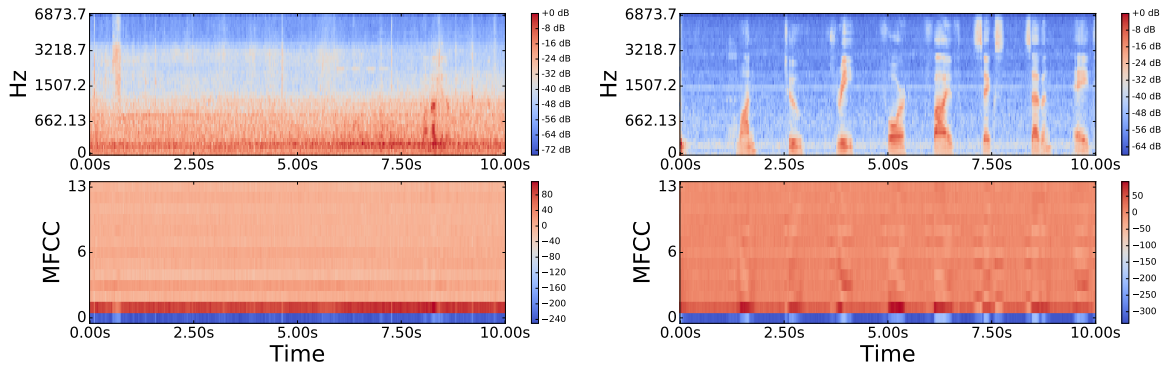


Figure 7: The Mel-power spectrum and the MFCC for the stationary context of a bus (a) and a speech (b) excerpts.

2.4.2 *Low-level audio features*

Many systems [53, 59] exploit low-level temporal, spectral and energy representations of the audio waveform to describe a scene. Among others, there are: zero-crossing rate expresses the number of sign changes in the signal; various energy-related descriptors (global loudness, root mean square power); auto-correlation features. There exist other low-level features extracted from spectral representation: the spectral centroid representing the centre of mass of the spectrum; the spectral flatness measuring the noise-likelihood of a sound; spectral roll-off indicating the frequency below which a certain percentage of magnitude distribution is concentrated. Low-level features are usually combined with MFCCs providing better performance than MFCCs alone.

A very specific feature for ASC integrates the magnitude of the power spectrum combining several frequency bands. This descriptor is independent from an absolute level of energy, because each energy sub-band is then divided by the total amount of energy. This feature, referred to as band energy ratio (BER), has been proposed for ASC by [38] and was used in various DCASE 2013 systems [53, 54, 59].

A complete list of these low-level audio features is detailed in [60].

2.4.3 *Spectro-temporal features*

Complex representations of audio signals aim at correlating spectral and temporal information in a compact way. These *time-frequency* features may have different forms. In [61], the authors mimic the behaviour of the mammalian auditory system. The signal is first filtered through a bank of 128 logarithmically scaled filters. Then the resulting filtered signal is integrated over a small window to capture fast variation of the signal over time. Similarly, the work in [62] computes the Mel-frequency spectrogram while compressing the amplitude with a log scale. Features in [55] are based on a model of the human cochlea.

2.4.4 *Spatial features*

Some acoustic scenes can be distinguished only by their two-channel stereo information. As an example, let us consider *tube* and *tubestation* for the DCASE 2013 dataset. The acoustic characteristics of these two classes are very similar. Nevertheless, the spatial information contained in the two-channel stereo wave may provide some cues: while for *tube* scenes the microphone is placed on the train, the *tubestation* can be identified by the passing of train noise from one channel to the other.

Spatial features employ a model for binaural hearing to estimate the interaural time difference (i. e. the difference in arrival time of a sound between left and right channels) and the interaural level difference (i. e. variation in amplitude of the two channels). The interaural coherence (i. e. a measure of the similarity between the reverberation received by each of the two channels) is usually added to complement the previous measures. All these three measures are computed for frames of 20ms and then integrated over the entire audio recording using statistics (mean, standard deviation) [54].

2.4.5 *Recurrent pattern features*

Most ASC approaches extract features from frames of 40ms overlapped by 20ms. These frame-level features are then compacted over longer period of time by using statistics (mean, standard deviation). According to [56], these statistic operations destroy the temporal recurrence of frame-level features. Recurrent information captures similarity between consecutive features (i. e. a stationary noise) or between periodic features (i. e. a repeated or semi-repeated sound).

The recurrence quantification analysis (RQA) feature extraction is composed of two parts. First, a 2-D similarity matrix is computed from adjacent MFCCs frames. This similarity is expressed as cosine distance between MFCCs vectors. The final binary matrix is obtained by thresholding cosine distances to a certain radius. The radius expresses the maximum distance between two non-consecutive frames which are still considered part of the recurrent series. Second, RQA is then used to extract compact metrics from the number, duration, and strength of elements in the similarity binary matrix. The authors propose then to average RQA metrics over the entire file duration.

2.4.6 *Image processing features*

The work in [33] makes use of image processing techniques to encode 2-D information. The authors propose to apply a histogram of gradients (HOG) approach on a time-frequency representation. The peculiarity of HOG is to capture the local directions of variation in a spectro-temporal representation. Let us imagine an engine noise which is accelerating or decelerating, or a scene composed of sound impulses. All these patterns are visible in a time-frequency representation with a specific local direction: diagonal in the case of accelerating/deceleration, vertical in the case of impulsive sounds or horizontal for stationary noises. The HOG features represent a whole image with the contributions of each single local direction.

The pipeline of a HOG feature extraction algorithm is summarised herein. First, the constant-Q transform is applied to an audio signal. In contrast to FFT, where the frequency and time resolution are fixed, the constant-Q transform has a better spectral resolution at lower frequencies and has a better temporal resolution at higher frequencies. The resulting time-frequency image is resized and smoothed to reduce strong variations and to make this representation as close as possible to a grey-scale image. This operation is called *pooling*. Different pooling strategies are proposed in order to reduce the dimensionality of the image while keeping the main orientations intact. The pooling has the advantages of augmenting HOG feature robustness to small translations in time or frequency which may affect the generalization on unseen data. Finally, this 'image' is split into non-overlapping cells and counting the gradient orientations in a given cell.

2.4.7 *Acoustic elements*

Some works in ASC literature describe an acoustic scene as a combination of simpler elements (i. e. audio events). Under this view, each acoustic scene can be represented as a histogram of detected acoustic events [63]. During the training phase, each context is modelled with a histogram of annotated audio events. During testing, an unknown event is first detected, classified and its occurrence added to this histogram. The histogram built during testing is then compared to that of each context to decide the most likely class. A similar approach has been proposed by [64] where the so called *acoustic unit descriptor* becomes the basic element of a more complex scene.

2.5 FEATURE POST-PROCESSING

After extracting a feature vector from raw data, feature post-processing extracts new information from an already-computed set of features. Post-processed features can be used as a replacement or in combination to the original features. In line with other classification tasks, here also feature post-processing techniques have been applied to ASC. Examples for DCASE 2013 include:

- principal component analysis (PCA), broadly adopted in many ASC systems [46, 61, 62] as a method to reduce dimensions while keeping intact the original variance of the feature space. PCA finds the best orthonormal bases according to different criteria. First, the variance of values projected onto this new set of bases should be maximal; second, the reconstruction error of the original and reconstructed space should be minimal. By optimizing these two criteria at the same time, the system will learn this set of orthonormal bases (called principal components). Hence, this set of bases are used to project the original high-dimensional feature space into a lower dimensional space where variance is maximal;
- Fisher score, which estimates the relevance of features by measuring how much features of a class are far to features of the other classes. A higher score is equivalent to likely separable features. With ASC being in a exploratory phase, Fisher score is used as metric to select the most discriminative subset of features [54];
- temporal derivatives over local frames, which capture the dynamic evolution of consecutive features. The most cited example of temporal derivatives are the Δ and $\Delta\Delta$ on the MFCCs [53, 54, 57] representing velocity and acceleration of features across time.

2.6 CLASSIFIER

After feature extraction and post-processing blocks, the original audio data is now represented in a new, meaningful space. A statistical parametric model learns from training samples how to classify new samples. Terms as *statistical* and *parametric* perfectly define the characteristics of model learning: *statistical* because the data available is a representative subset of the entire population and the model has to be generic enough to classify unseen data; *parametric* because the classifier itself is defined with a set of parameters (e. g. the mean, standard deviation and weigths of a GMM). During training, a parametric function defining the model is optimized. During testing, a decision function assigns an unknown sample to the most likely class.

The ASC literature is characterised by two types of model: generative models which learn the underlying distribution of the training samples and discriminative models which learn a decision boundary between classes.

2.6.1 *Generative models*

From observing scene-specific samples (e. g. *car*, *bus*), we can build a model of what a scene sounds like. During testing, a new sample z is compared to the different models to predict the most *likely* one.

It is supposed that a new sample has been generated by an underlying distribution $\Pr(z|\theta_c)$. This is a *likelihood* which measures how likely it is that the model parameters θ_c of class c generated sample z.

$$\Pr(\theta_c|z) = \frac{\overbrace{\Pr(z|\theta_c)}^{\text{likelihood}}\overbrace{\Pr(\theta_c)}^{\text{prior}}}{\Pr(z)}. \tag{4}$$

$\Pr(\theta_c)$ is referred to as *prior* probability. It indicates the knowledge about the how likely it is to encounter the c^{th} class. $\Pr(z)$ is a term common to all C classes and it corresponds to $\Pr(z) = \sum_{c=1}^{C} \Pr(z|\theta_c)\Pr(\theta_c)$. When calculating $\Pr(\theta_c|z)$ to predict a new sample, the denominator can be omitted since

$$\arg\max_c \frac{\Pr(z|\theta_c)\Pr(\theta_c)}{\Pr(z)} = \arg\max_c \Pr(z|\theta_c)P(\theta_c). \tag{5}$$

While learning a generative model, *likelihood* probabilities are not known in advance and they have to be estimated on training data. For some classification problems, *prior* probabilities are estimated on the class distribution of the training; for other problems *prior* probabilities are given. The GMM classifier is an well known example of a generative classifier utilised as baseline for DCASE 2013 [6]: the learned model, in this case, indicates how likely a sample has been generated by a multivariate Gaussian distribution.

As for MFCCs, the choice of this classifier has been adopted as reference classifier for ASC [38, 39]. Intuitively, the acoustic space of an audio scene can be modelled as a multi-modal density distribution where each component may represent a spectral related hidden class (i. e. a *bus* scene is composed by components such as engine, tires, door opening). In the case of GMM, distribution of training samples X_c of class c is modelled as a weighted combination of Gaussian distributions:

$$X_c \sim \prod_{i=1}^{K} w_i \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{6}$$

where K is the number of components, i^{th} expresses the current components, $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is the Gaussian distribution with mean $\boldsymbol{\mu}_i \in \mathbb{R}^D$ and covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{D\times D}$ (with D as the dimensions of feature vectors). Thus, each sample in X_c is generated by a weighted mixture of these K Gaussian distributions. The probability to belong to a specific mixture i depends on the latent variable w_i. $\theta_c = \{w_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1\cdots K}$ expresses the model parameters for class c and they are learned during the training phase.

In fact, finding the maximum likelihood of θ_c is not easy because mainly depends on the distribution of latent variable w_i. These parameters are obtained trough an iterative process called expectation-maximization (EM) algorithm [65].

Once models θ_c for all classes $c = 1 \cdots C$ have been inferred from the training data, the same processing and feature extraction are applied to an unlabelled audio sample. This new sample z is then evaluated using models from all classes C in order to determine the most likely class $\hat{c}$

$$\hat{c} = \arg\max_c \Pr(z|\theta_c)\Pr(\theta_c), \quad c = 1, \cdots C. \tag{7}$$

In the particular case of acoustic scene classification, GMM baseline classifier has some peculiar aspects:

- the decision of the most probable class $\hat{c}$ depends on the sum of the log likelihood score integrated over a longer sequence of features (about 30s);
- the ordering of this sequence is not taken into consideration. Any random permutation of the features relatively to the recording would produce the exact same prediction;
- the choice of the number of Gaussian components K is critical. Values of K have to be big enough to well represent the multi-modal distribution of samples of an acoustic scene. On the contrary, a number of components too big may overfit the training data by learning a too precise distribution of training data.

Other classifiers take advantages of GMM modelling by extending temporal evolution. The temporal evolution of sounds has been pointed as one of key for classifying a scene [27, 39, 59]. For example, the sequence as "unlocking the car - opening of the door-engine starting" would likely identify *entering to a car* . Based on sequence of GMM densities, HMM models the probability that a sound occurs after another. These probability compose the HMM transition matrix which contains the transition probability between sounds at different times. If the previous three sounds (unlock, door and engine) occur one after the other, the HMM will return a higher probability while a different order will generate a lower probability.

Another method exploiting generative models is the i-vector system, initially adopted in speaker verification community to separate the speaker characteristics from non-relevant information (such that channel variations, acoustic environment, etc.). First, all GMM parameters are concatenated in a high dimensional vector. Then a lower-dimensional vector (referred to as i-vector) is finally generated from the original high dimensional vector. An i-vector represents the identity of an acoustic scene [57] without any spurious information (e. g. channel variation).

2.6.2 *Discriminative models*

As for generative models, the goal of discriminative models is to assign a sample z to one of the classes. The main difference is that this probability learns directly the function which maps from the feature space of z to class c. The simplest example is the logistic regression, where this *mapping* function is defined as $f_\theta = s(\boldsymbol{\theta}^\top z)$ with s being the sigmoid function, and $\boldsymbol{\theta}$ the mapping which minimizes the error between real classes and predictions.

Classifiers included in this group are the support vector machine (SVM), which learns how to separate features of different classes with an hyperplane and the k-nearest neighbours (KNN) approach, which assigns to the class whose its k nearest neighbours belong to.

Due to its simplicity, kNN has been used in early ASC works [36] because complex aspects of the data can be learned by local approximation. As drawbacks, kNN heavily depends on the number of neighbours and the quantity of data available. SVM, instead, projects original

features into a higher-dimensional space in which scenes may be linearly separable. This is achieved according to an hyperplane which maximises the margin between classes, thereby minimising classification error.

In the group of discriminative classifiers, decision trees (DTs) are examples of non-parametric supervised methods which learn the decision boundary by recursively partitioning the space. Since DTs can create over-complex structures and lead to unstable classifiers (small variation in the data might result in a different structure of the tree and therefore a bad generalization), this problem is mitigated by using DTs within an ensemble of decision tree classifiers.

The tree-bagger classifier employed in [58, 66] is a possible variant of ensemble learning, where the final model is an aggregation of decision trees trained on an independent subset of the training data. The final prediction consists of averaging the ensemble of all DTs predictions.

It is worth to underline that in 2013 deep neural network techniques (which were already showing improvements in speech and image recognition tasks) were rarely mentioned. Unfortunately, the amount of data in the DCASE 2013 database was not sufficient to apply deep learning approaches with success.

The SVM is perhaps the most popular classifier among DCASE 2013 systems (6 out of 11). The reason of this success is linked to the peculiarities of SVM compared to other methods:

- the kernel trick allows SVMs to transform the original feature space in a higher dimensional space where separation is linear (i. e. it introduces non-linearity mapping in the feature space);
- it is defined by a convex optimisation problem for which very efficient methods exist, speeding up the learning phase;
- it is more efficient than generative models when the quantity of data is limited and the number of classes is high (which is exactly the case of DCASE challenge).

2.7 TESTING STRATEGIES

After model learning and feature transformation, several strategies exist to decide the class of an unknown sample. Whereas some of the strategies are highly connected to a specific type of classifier, others can be applied universally. When the classifier is discriminative and able to separate only two classes (as SVM), multiple classifiers can be combined to discriminate between C classes. In the *one-versus-all* combination, C binary classifiers are learned and then tested over the remaining $C-1$ models selecting the class with the highest margin. In *one-versus-one*, instead, $\frac{C(C-1)}{2}$ possible paired combinations of binary classifiers are learned. During testing, the predicted class is the class which has been selected by the majority of binary classifiers.

When the original audio recording is split into shorter segments, it may be difficult to provide a single prediction of each recording. Majority vote is a testing strategy which assigns the most occurring scene as final prediction. An example in [53] assigns as final prediction the class which has occurred the most across all 4s segments composing a 30s recording.

2.8 DCASE 2013 RESULTS

This section summarises the results of works presented in previous sections. The comparison of these systems guided the research in the next chapters and it was used as a common

Method (ID)	Features	Classifiers	Testing strategies
Olivetti et al. (OE) [66]	Length of the compressed audio file	Random forest based on the compression distance	
Elizalde et al. (ELF) [57]	MFCCs + Δ + $\Delta\Delta$ over a concatenation of left, right, difference and average of stereo channels	GMM-UBM $\rightarrow$ i-vector	Maximum likelihood
Krijinders et a. (KH) [55]	time-frequency chocleagram	statistics $\rightarrow$ SVM	One-vs-one
Baseline	MFCCs	GMM	Maximum likelihood
Patil et al. (PE) [61]	time-frequency multi-resolution analysis $\rightarrow$ PCA	SVM	One-vs-one, weighted majority vote by the energy present in 1s window (overlap 0.5s)
Nogueira et al. (NR) [54]	MFCCs, temporal features (modulation rate of MFCCs over 4 bands, event density estimation), spatial features (time and amplitude differences between the two channels) $\rightarrow$ Fisher score for features selection	SVM	
Nam et al. (NHL) [62]	unsupervised learning using restricted Boltzman machines on Mel-spectrogram $\rightarrow$ PCA	SVM	One-vs-all
Chum et al. (CHR) [59]	energy/frequency features over short and long frames (different temporal resolutions)	GMM $\rightarrow$ HMM	Maximum likelihood
Geiger et al. (GSR) [53]	spectral, cepstral, energy, voicing-related over 4s of signal	SVM	Majority vote
Rakotoma-monjy et al. (RG) [33]	Histogram of gradients on constant Q transforms	SVM	One-vs-one
Li et al. (LTT) [58]	MFCCs on wavelet decomposition	Ensemble of binary trees	Majority vote
Roma et al. (RNH) [56]	MFCCs $\rightarrow$ recurrent quantification analysis metrics (RQA)	SVM	One-vs-one

Table 1: The list of the systems submitted to DCASE 2013 challenge, followed by the type of features, classifier and testing strategies. The arrow expresses dependencies from *feature* $\rightarrow$ *feature processing* or *classification* $\rightarrow$ *meta-classification*. A white space indicates that the information is not provided or specified.

reference. Herein, systems submitted to DCASE 2013 challenge are listed in Tab. 1 according to the type of features, classifiers, testing strategies as discussed in Sec. 2.4 and 2.6.

Performance evaluation regards the accuracy averaged over a 5-fold cross validation. In the specific case of DCASE 2013 challenge, each system has been trained with 8 audio files for 10 classes and tested on the remaining 20. This has been repeated by the number of folds. Both development and evaluation sets use a 5-fold validation to train and test performance, optimizing on the small set of available data.

Results for development and evaluation sets are reported in Fig. 8: full blue circles represent accuracy of the evaluation set averaged over 5-fold; the "x"s in red indicate the accuracy for the development set; the bars illustrate the confidence intervals of the mean accuracies in the evaluation set (not all confidence intervals were reported for the development set). A confidence interval (CI) measures the distance of the mean calculated on 5-fold (sub-samples of the population) from the mean calculated on an infinite number of folds (the entire population). The values of CI are found by multiplying the 95% quantile of a standard normal distribution $q^{0.95}_{\mathcal{N}(0,1)} = 1.96$ with the standard error: $\frac{\sigma}{\sqrt{5}}$. The upper and lower bounds of the CI are then added or subtracted from the mean accuracy $\mu \pm 1.96\frac{\sigma}{\sqrt{5}}$.

CIs express the 95% of probability of having the true expected value of the accuracy in this range. As a consequence, CIs estimate the range in which the real accuracy lies, considering the variation of the metric in each fold. Intuitively smaller datasets mean wider CIs (and higher uncertainty).

2.8.1 *Discussion*

Fig. 8 shows significant difference between results for the evaluation and development set performance. Abbreviation of the submitted works are reported in Tab. 1.

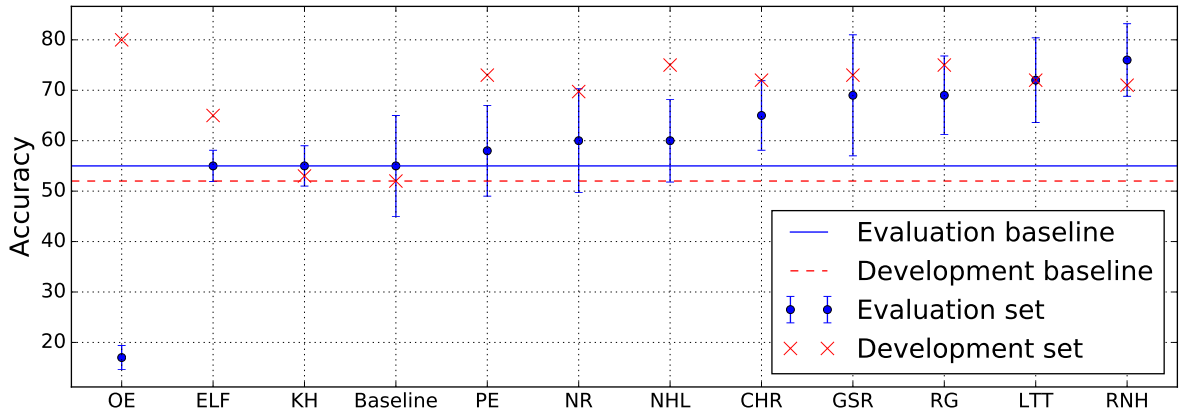


Figure 8: Plot shows the accuracy mean with 95% confidence intervals (CI) over 5-fold cross-validation for DCASE 2013 dataset. In blue circles the values of evaluation set, whose baseline is expressed also with a blue line; in red stars the values of the development set with the baseline expressed in dashed red line. For some systems, the CI are not provided in the description of the development set and there were not reported.

Some methods were probably overfitted to development data. In general, the best systems (LTT, RNH) improved performance for the evaluation set. A significant number of systems perform better than the baseline and even in the case of similar accuracy, ELF or KH systems should be preferred for a lower CI.

Hence, performance and methods are highly correlated. Except for ELF and CHR systems, all the others employ discriminative classifiers (SVM, Binary tree). On the feature side, MFCCs are the most adopted. Among several testing strategies, the majority vote seems to be the most effective allowing to integrate decisions across time. This suggests that an acoustic scene is reliably detected at 30s, as found in [38].

Due to its broad adoption by many submitted systems, SVM does not make the difference in term of final performance. Indeed, by analysing the best three systems, both RG and RNH systems propose ASC-tailored features: the former by capturing temporal structures using CQT-based images representation; the latter by quantifying recurrence of consecutive MFCCs. The idea of exploiting time-frequency spectrograms is common to other systems (KH, PE, NHL, LTT) suggesting that temporal information is relevant to the ASC task.

From a global point of view, the fact that only few systems outperform the baseline prove the difficulty of the task for a modest amount of data. Moreover, it seems that a similar level of performance obtained in other domains (such as speech recognition or music genre classification) could be achieved from a deeper investigation on ASC-tailored features.

2.8.2 *Conclusions*

This chapter described the literature in ASC from the early works in 1997 until latest state-of-the-art systems in 2013, connecting psychoacoustic studies to computational methods. There exist some common trends in ASC literature. They are detailed in the following:

- ASC is a difficult task for both humans and machines, confirmed by subjective tests on 19 individuals [37]. When humans try to classify a scene, other cues (e. g. visual) are fused with audio to complete it with missing information;
- temporal integration has been proven effective by both cognitive studies and computational testing strategies. Majority vote is an example of integrating fixed-length segments over 30-40s of signal. The analysis of the scene through short segments can

capture prominent sounds (useful to discriminate a scene) and, at the same time, may remove the less informative clips (e. g. silences);

- interpretations of different methods support the intuition that there exist for ASC two main approaches. A top-down approach starts from general characteristics to gradually recognising peculiar sounds (e. g. hierarchical classifier starting from indoor/outdoor or transport/non-transport mode). A bottom-up approach instead builds its classification on audio patterns which build the entire scene (e. g. the occurrence of certain audio events can identify a scene);

- the most accurate systems designed specific features tailored to ASC problem. These systems capture time-frequency evolution or recurrence of features. Another important aspect to consider is that most systems employ a standard SVM classifier, so the difference in term of performance is related to the features;

- the introduction of a publicly shared dataset has boosted research in this domain, allowing a fair comparison of different systems. Nevertheless, the small amount of data and the low variability means that the DCASE 2013 dataset is still too far from real conditions (i. e. different microphones and channel paths). Furthermore, some methods (such as deep neural networks, which are known to require a significant amount of data to be trained) couldn't be tested.

Finally, ASC is a recent and exciting area of research with many possible directions. One of them is to investigate features and aspects specific to scene classification. In that sense, the analysis of ASC features will occupy a large portion of this thesis. The second trend concerns the application of ASC on real devices and this is partially missing in the current literature. A real-time and low-complexity ASC system requires different approaches compared to current methods. Contributions in this area will be treated in the Part 2 of the manuscript.

3

A STATE-OF-THE-ART SYSTEM AND LIMITATIONS

This chapter details the state-of-the-art implementation of an ASC system. It is used in the remainder of this thesis as a reference system. It is based upon the winning system of DCASE 2013 (referred to as Roma et al. system (RNH) [56]).

Several aspects of the RNH system are presented in this chapter: feature extraction and post-processing (in Sec. 3.1); some insights about the SVM classifier (in Sec. 3.2); the impact of parameters on final performances (for both features and classifier in Sec. 3.4); and finally, a discussion about the main limitations of this system (in Sec. 3.5).

3.1 RNH FEATURE EXTRACTION AND POST-PROCESSING

The RNH system employs a form of feature extraction which captures recurrent frame-level features over a period of time. Frame-level features are MFCCs, extracted from every frame with a 50% overlap. The processing time window adopted in RNH is 400ms (containing 40 MFCCs computed every 25ms overlapped by 15ms). These 40 MFCCs are compared to each other to obtain a similarity matrix between each pair of 40 MFCCs. In this case, *similarity* means the cosine distance computed between MFCCs vectors. The matrix of cosine distances is then thresholded with a radius r to obtain a binary similarity matrix.

Fig. 9 shows an example of such a binary similarity matrix computed from 40 consecutive MFCCs of a bus scene. Ones and zeros indicate similar and non-similar MFCC vectors respectively. According to this view, a diagonal lines indicate consecutive *periodicities* of MFCCs while vertical lines depict *stationarities* (similar MFCCs on consecutive frames). The diagonal lines of the binary similarity matrix correspond of identical MFCC vectors and thus not informative.

From the analysis of this matrix, several measures are derived to quantify the length and direction of points of the matrix. These measures are essentially statistics over the thresholded matrix [67]. Examples include the *recurrence* (RR), the percentage of ones in the matrix, the *determinism* (DET), percentage of points lying in diagonal lines, or the *laminarity* (LAM), the percentage of points on vertical lines.

These measures quantify the temporal recurrence of the acoustic scenes. For example, a *bus* noise characterised by stationary sounds (e. g. engine noise, tires, etc.) has a higher percentage of vertical lines; a restaurant identified with a weaker temporal structure has a lower percentage of ones in the matrix.

The full set of recurrent quantification analysis (RQA) features extracted over 40 consecutive MFCCs refers to a long window of 400ms, while MFCCs represent shorter frames. In the RNH system a combined feature vector of MFCC and RQA is obtained by temporally averaging RQA measures and computing their mean and standard deviation over time.

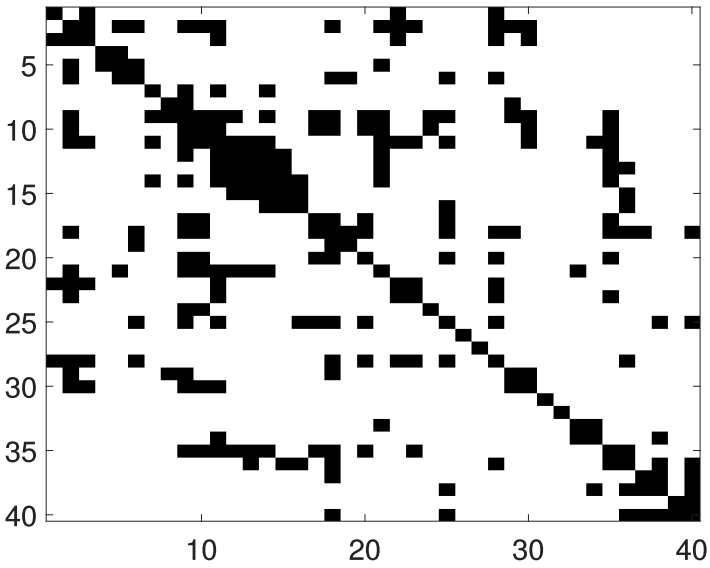


Figure 9: Recurrence plot after thresholding the similarity matrix of 40 consecutive MFCCs. The cosine similarity provides the values which are then thresholded. The black bins describe the frames considered similar. Except on the diagonal (where the self-similarity is not relevant), all lines and diagonals reflect periodicities and stationarities of MFCC frames. This excerpt belongs to an example of bus from DCASE 2013.

3.2 SUPPORT VECTOR MACHINES

The SVM is the classifier used in the RNH system. The core idea of the SVM is to project data into a higher dimensional space in which a linear separation is possible. The main reasons for its popularity in machine learning can be summarised as follows: i) the separation is formulated as quadratic convex problem whose solution is unique; ii) there exist fast and efficient methods to solve the quadratic problem; iii) for small datasets the discriminative nature of the SVM provides higher performance than generative classifiers (which would require more data to be trained).

3.2.1 *The margin*

The goal of the SVM classifier is to find a decision boundary for which the average distance between the training points and the boundary is maximised. The maximisation is achieved by first computing the margin m_n between each training sample $(\boldsymbol{x}_n, y_n)$ and the boundary

$$m_n = y_n(\boldsymbol{w}^T \boldsymbol{x}_n + b), \tag{8}$$

where y_n is the class label and $\boldsymbol{w}, b$ are the parameters of the boundary. Intuitively, $\boldsymbol{w}^T \boldsymbol{x}_n + b$ has to be positive when $y_n = 1$ and negative number when $y_n = -1$. The confidence in the classification is directly proportional to maximum distance between the samples $\boldsymbol{x}$ and the boundary (expressed with $\boldsymbol{w}, b$).

Given a training set of $(\boldsymbol{x}_n, y_n)$, the smallest margin m among all possible margins gives a measure of the separability between the two classes:

$$m = \min_{n=1,2,\cdots N} m_n, \tag{9}$$

where m_n is the margin computed from the n^{th} training sample $(\boldsymbol{x}_n, y_n)$.

3.2.2 *Optimal margin classifier*

A parametric optimization function maximises m with respect to $\boldsymbol{w}$ and b

$$\begin{aligned} &\max_{\boldsymbol{w},b} \quad m \\ &\text{s.t.} \quad y_n(\boldsymbol{w}^T \boldsymbol{x}_n + b) \geqslant m, \quad n = 1, 2, \cdots N \\ &\qquad \|\boldsymbol{w}\| = 1 \end{aligned} \tag{10}$$

The constraint $\|\boldsymbol{w}\| = 1$ ensures the scalability invariance of the margin. It is a non-convex function which makes any optimization routine difficult to apply. Nevertheless, some operations can transform this problem into a solvable one without changing its nature. First, the constraint $\|\boldsymbol{w}\| = 1$ can be nested directly in the function by dividing m by $\|\boldsymbol{w}\|$. As a second step, we can introduce a scaling factor that forces $\boldsymbol{w}, b$ to produce a margin equal to 1. As a result, since the inequality $y_n(\boldsymbol{w}^T \boldsymbol{x}_n + b) \geqslant m$ holds for both sides of the margin ($y_i \pm 1$), the margin distance is doubled. This is due to the symmetry on both positive and negative sides of the margin itself. The optimization problem is now given by

$$\begin{aligned} &\max_{\boldsymbol{w},b} \frac{2}{\|\boldsymbol{w}\|} \\ &\text{s.t.} \quad y_n(\boldsymbol{w}^T \boldsymbol{x}_n + b) \geqslant 1, \quad n = 1, 2, \cdots N. \end{aligned} \tag{11}$$

Note that maximizing $2/\|\boldsymbol{w}\|$ is equivalent to minimizing $\|\boldsymbol{w}\|/2$. In addition, since the quadratic form is a strictly decreasing function, minimizing $\|\boldsymbol{w}\|/2$ will provide the same minimum of $\|w\|^2/2$, with the difference that the latter is differentiable and better suited to optimization:

$$\begin{aligned} &\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 \\ &\text{s.t.} \quad y_n(\boldsymbol{w}^T \boldsymbol{x}_n + b) \geqslant 1, \quad 1, 2, \cdots n \end{aligned} \tag{12}$$

Eq. 12 expresses a convex quadratic objective problem with linear constraints and it is practical for two reasons:

- efficient quadratic programming codes are widely available to solve this kind of problem;
- the solution to the optimization function is found through an iterative algorithm [68].

Eq. 12 can be rewritten with a well known optimisation technique (called duality or dual formulation) and methods such as Lagrangian multipliers are used to solve it. Seeing the problem in a another perspective (the dual formulation) provides a lower bound to the solution of the primal problem. In general, the optimal values of the dual problem are not the same of the primal and their difference is called the *duality gap*. However, under certain conditions, we may solve indistinctly the dual problem for the primal. These conditions are referred to as the Karush-Kuhn-Tucker (KKT) conditions and they determine whether optimal values of the primal problem are equal to the optimal of the dual problem. Details of the dual problem formulation are presented in Annex A.1.

3.2.3 *Soft-margin and C parameter*

There exist real cases (ASC is one good example) where the two classes are not *perfectly* separable and boundary is affected significantly by outliers in the training data. At the same

time, errors (i.e. samples on the wrong side of the boundary) have to be minimized while keeping the margin as large as possible:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_n \xi_n \\ \text{s.t.} \quad & y_n(w^T x_n + b) \geqslant 1 - \xi_n, \\ & \xi_i \geqslant 0 \end{aligned} \tag{13}$$

where ξ_i indicates the slack variable, C expresses a trade-off between a smaller $\|w\|^2$ (which corresponds to a large margin) and a small amount of training samples having $\xi_n = 0$. In other words, the C parameter is the cost of misclassifying training samples. For large values of C, the optimization will choose a smaller-margin if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that boundary wrongly classifies more points. The dual problem is found to be similar to that in Eq. 50, the only exception being the α constraints:

$$\begin{aligned} \max_{\alpha} \quad & \sum_n \alpha_n - \frac{1}{2}\sum_{n,m} \alpha_n \alpha_m y_n y_m x_n x_m \\ \text{s.t.} \quad & \sum_n \alpha_n y_n = 0 \\ & 0 < \alpha_n < C. \end{aligned} \tag{14}$$

KKT conditions determine the training samples which "support" the final classification. These *special* samples are called support vectors (SVs). Due to KKT conditions, each training samples x_n assumes a meaning depending on the value of α: for $\alpha_n = 0$ or $\alpha_n = C$, the corresponding sample x_n is not a SV; for $0 < \alpha_n < C$, x_n becomes a SV.

3.2.4 *The Kernel trick*

In the dual formulation (Eq. 50), every inner product $x_n x_m$ can be replaced by a kernel function $K(x_n, x_m)$ to have a more powerful representation. The idea is to map the data into a higher dimensional space where the boundary is optimal in separating two classes. Instead of calculating the new coordinates in this space for all features, the inner products between all pairs of data are calculated. One of the most popular choices is the Gaussian kernel, expressed as:

$$K(x_n, x_m) = \exp\left(-\frac{\|x_n - x_m\|^2}{2\sigma^2}\right). \tag{15}$$

When two samples are really close (i.e. $x_i \simeq x_j$) then $K \to 1$. In contrast, when two samples are far (i.e. $x_i \neq x_j$), then $K \to 0$.

The variance σ amplifies the distance between x_n and x_m. A graphical illustration of the influence of σ on a toy-example classification problem is depicted in Fig. 10. If the distance between them is much larger than σ, the kernel function tends to be zero. Thus, if σ is very small, it will have a small influence on the distance. In other words, a higher value of σ will have a smoother decision boundary with a risk of underfitting; a lower σ will have a finer boundary with a greater risk of overfitting (the decision function is much more complex).

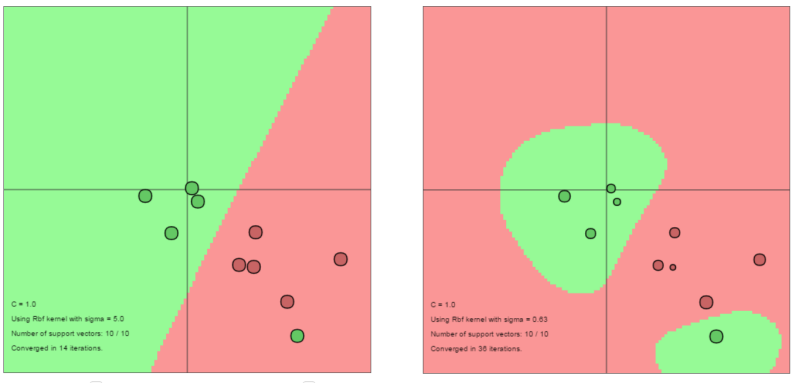


Figure 10: The effect of σ for the gaussian kernel: on the left side of the figure is represented a smoother decision boundary produced with $\sigma = 5$; on the right the sharper decision boundary produced by $\sigma = 0.63$.

3.2.5 *Normalisation*

The feature normalisation ensures that all dimensions in the feature vector have the same range [69]. First the mean and the standard deviation of each dimension are computed from the training set. Then, each sample in the training, validation and testing sets is scaled by subtracting the mean and dividing by the standard deviation. This procedure is referred to as *z-score*. With this operation, all the features in the training are scaled to have zero mean and unit variance and the same normalisation is applied subsequently to the validation/testing samples. The main reason of normalising is to avoid features with larger values influencing widely the decision criteria at the expense of features with smaller values.

3.2.6 *Grid-search strategies*

Standard SVM classifiers require the combined tuning of two free parameters: the trade-off between the error/margin C and the width of the Gaussian kernel σ. The parameter σ, as mentioned in Sec. 3.2.4, amplifies or smooths the distances within the Gaussian kernel. As a consequence, at higher values of σ correspond simpler boundaries (see Fig. 10).

Grid-search routines allow the testing of several combinations of (C, σ) in order to select the pair whose accuracy is the highest as judged on a training sub-set (referred to as *validation set*).

As illustrated in Fig. 11, the grid-search performed on the first fold of the DCASE 2013 development set follows two criteria: on the right of the figure, the pair is selected accordingly to the highest validation accuracy; on the left of the figure, the ratio between the number of SVs and the training size is considered instead. Let us call this ratio *SV ratio*.

SV ratio expresses the generalisation capacity of the SVM classifier: when SV ratio is 1, SVM uses all training samples as support vectors; when SV ratio goes to 0, few training samples become effectively support vectors. According to Vapnik studies [70], less complex models are less likely to overfit. Hence, SV ratio directly relates to the classifier complexity.

In the right inset of Fig. 11, the SV ratio reaches a value of 1 for every pair of (C, σ) parameters, showing that indeed SVM needs to use almost every single training sample to create an optimal boundary. The classifier complexity information is not captured by the accuracy-only criterion (left inset of Fig. 11).

Herein there are some comments about two grid-search criteria (Fig. 11), one based on the best accuracy on the validation set and the other on the lowest SV ratio:

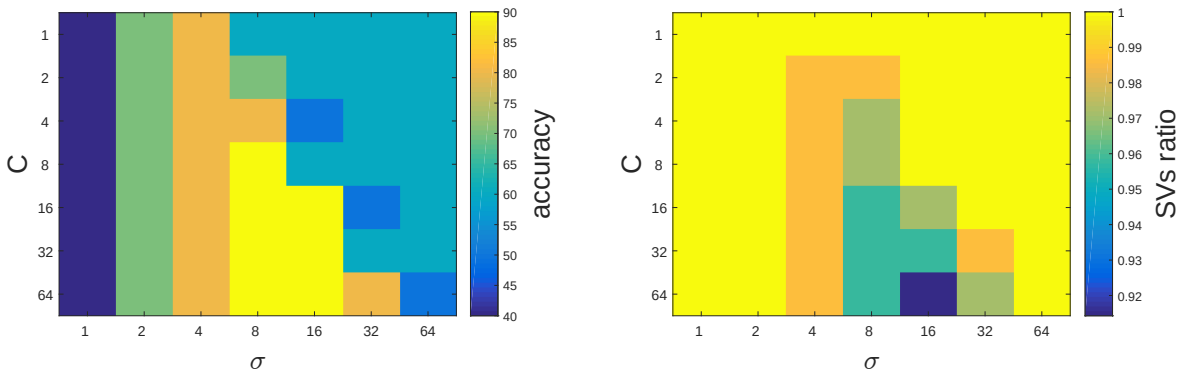


Figure 11: Visualization of two different strategies of grid-search: on the left the accuracy for each pair of C, σ is reported; on the right the ratio between the number of SVs and the total size of the training.

- both grid-search strategies suggest the same pair of C, γ. Interestingly the SV ratio, while not relying on any validation data, provides a pair of C, σ which corresponds to the most accurate on the validation set;
- there exist C, σ whose models produce the same accuracy. For example, pairs as $C = [64, 32, 16]$ and $\sigma = [8, 16]$ have a 90% accuracy on the validation. By focusing only on accuracy, all these pairs are equivalent. The SV ratio, instead, adds the information of the classifier complexity. Under this vision, the best pair would be $(C = 64, \sigma = 16)$ which represents the highest accuracy and the lowest SV ratio;
- the lowest SV ratio is around 0.92 meaning that almost all training samples are SVs. This is probably caused by the limited amount of data available for training. The problem of the modest amount of data is discussed later in the thesis (sec. 4.3).

Therefore, a better tuning of (C, σ) should consider both accuracy and SV ratio as optimisation criterion. The best pair becomes the one minimizing the following criterion λ:

$$\lambda = \text{SV ratio} + (1 - \frac{\text{accuracy}}{100}), \tag{16}$$

where the accuracy (expressed as a percentage) takes the same range of SV ratio. The (C, σ) whose model generated the lowest λ are finally selected.

The two different strategies are compared in Fig. 12: one based on Eq. 16 (called SVs & accuracy), the other on the best validation accuracy (best accuracy). In order to fairly compare the two strategies on a different composition of training and validation sets, a bootstrapping test has been employed by random sampling the training set and validation set 20 times. The values of distribution obtained correspond to the final accuracy on DCASE 2013 evaluation set. The distribution of these accuracies is depicted with box-plots in Fig. 12. Box-plots show the first quartile to the third quartile range (solid box) and the min-max interval (black line). Median is depicted with a horizontal red line, mean with a red square.

The proposed grid-search strategy report a lower quartile range, while the accuracy-based has values that spans 74% and 78%. Global accuracy passes from 76% for the accuracy-only strategy to 77% for the proposed grid-search. The proposed grid-search strategy λ produces a simpler model (with fewer SVs) and is therefore less prone to overfitting. This grid-search strategy is adopted for all experimental works reported later in this thesis.

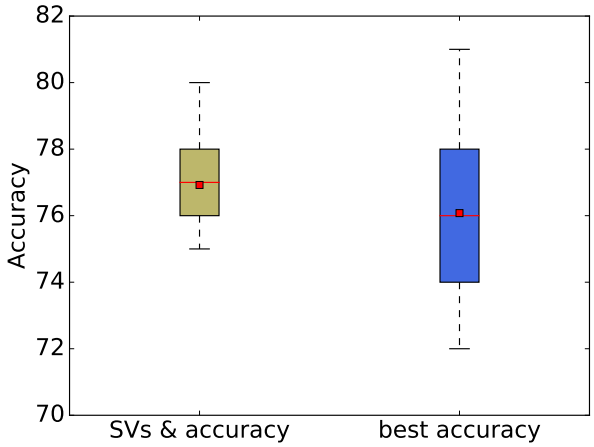


Figure 12: Box-plots representing the accuracy distribution for the two grid-search strategies: the one on left is the proposed grid-search which takes into account the validation accuracy and SV ratio; on the right the accuracy-only strategy.

3.3 THE STATE-OF-THE-ART SYSTEM RE-IMPLEMENTATION

Similar to other ASC systems, the RNH system follows a standard approach: audio preprocessing, feature extraction, feature post-processing, classification and testing. As first part of the experimental work reported in this thesis, the RNH system was re-implemented in order to establish a baseline algorithm. Specific implementation details are reported in the following:

1. **pre-processing** involves all the processing done on the raw-audio waveform. Only the left channel of the stereo wave form is used for feature extraction;

2. **feature extraction** is based on MFCCs which are computed with the default settings of *rastamat* library [71]. The frequency range is set to $[0, 900]$Hz. 3000 MFCC feature vectors calculated from short windows of 25ms with a shift of 10ms;

3. **feature post-processing** is applied to the MFCC matrix (13 MFCC coefficients $\times$ 3000 frames). MFCCs statistics are computed from frame-level MFCCs resulting into a single 26 dimensional feature vector. The RQA features are computed over batches of 40 adjacent MFCCs (400ms of audio). RQA features are then averaged over time and added to the MFCC feature vector to form a 37-dimensional feature vector. The same operation is performed for each file. In contrast to the experiments reported in [56], the removal of the first MFCC coefficient (C0) did not improve on performance and it is therefore retained;

4. **classification** follows a standard SVM-based approach with a Gaussian kernel (radial basis function (RBF)). It was implemented with the well known *libsvm* library [72];

5. **testing** comprises a one-to-one approach, resulting in 45 possible paired class combinations. The class which *wins* the majority of paired class combination is selected as the most likely.

As a conclusion of this section, the performance of the RNH system and the implementation of same system are reported in Tab. 2. `MFCC+RQA-900` indicates the re-implemented system: the term `MFCC+RQA` refers to the feature extraction type whereas the term `900` to frequency range from which MFCCs are computed. The two systems have comparable performances: on DCASE 2013 development set, the difference in term of accuracy is 1%; on

	DCASE 2013 - dev	DCASE 2013 - eval
RNH [56]	71%	76% ±7.2
MFCC+RQA-900	70% ± 10.2	76% ±5.7

Table 2: The difference between the results reported in the literature for RNH and the re-implementation of the same approach done in this work. The results describe the average accuracy over 5-fold cross validation with their corresponding confidence interval.

DCASE 2013 evaluation set, the two systems achieve an accuracy of 76%. A Wilcoxon signed rank test [73] confirms that the differences for the DCASE 2013 (dev and eval sets) are not statistically significant. The provided results are inline to those reported in the literature. `MFCC+RQA-900` will be indicated as the state-of-the-art system for the experimental works reported later in this thesis.

3.4 LIMITATIONS OF THE CURRENT APPROACH

This section aims at showing how the feature tuning assumes a primary role in ASC. Albeit reporting the best performance on the DCASE public dataset, `MFCC+RQA-900` performance is stricly related to the DCASE 2013 class composition. Experiments on `MFCC+RQA-900` system confirm that small changes in feature tuning have a significant impact on final performance. The main limitations of the `MFCC+RQA-900` system are summarised as follows:

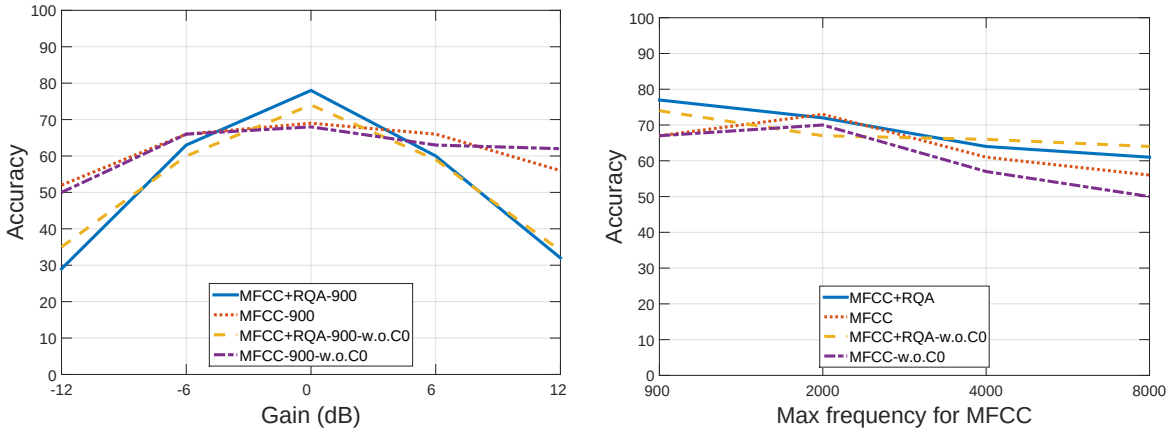
- energy-dependent features (Sec. 3.4.1). A difference in the energy level can change the feature values and, therefore, the final classification;
- 900Hz frequency-range (Sec. 3.4.2). While this range suits the DCASE 2013 class composition, it may not be optimal for other datasets with a different set of classes;
- 30s segment predictions (Sec. 3.4.3). `MFCC+RQA-900` calculates statistics over 30s segments only, but is not suited to shorter (or longer) segment durations;
- weak temporal structure (Sec. 3.4.3). By computing the MFCC and RQA statistics over the full recording, `MFCC+RQA-900` system loses the temporal information of the scene.

3.4.1 *The role of C0*

The computation of the first MFCC coefficient (C0) is linked to the first component of the DCT $i = 0$ for which the cosine function is equal to 1 (see Eq.3). This is equivalent to summing up all log-energies representing the overall loudness of the signal. This information improves the recognition performance of the DCASE 2013 database, but may lead to poor generalisation when a significant difference is encountered on unseen samples. This difference can happen for two main reasons:

- a gain is applied to the signal during the testing phase (i. e. some devices may introduce a gain on specific frequency bands);
- the distance of the microphone to the sound source changes the absolute sound level of the recording;

To simulate these effects, a set of experiments are performed with different amplifier gains being applied to testing samples of each fold while training samples are left unaltered.



(a) (b)

Figure 13: On the evaluation set of DCASE 2013, examples of feature tuning on 4 systems: (a) accuracy as a function of different dB gains applied to the testing samples (at 0dB no gain has been applied); (b) accuracy as a function of different frequency range used by MFCCs (the minimum frequency is fixed at 0Hz). The accuracies are computed on systems re-trained at each frequency range.

Amplifier gains g are expressed in dB as $[-12, -6, 0, 6, 12]$ where at 0dB no gain is applied. A gain factor of $10^{\frac{g}{20}}$ is applied to signal amplitudes.

Fig. 13 (a) illustrates recognition with and without C0, for `MFCC+RQA-900` and `MFCC+RQA-900-w.o.C0` systems respectively. For sake of completeness, results for systems without RQA features are also shown and referred to as `MFCC-900` and `MFCC+RQA-900-w.o.C0`. It is stressed that C0 is removed from the average only (it is kept in the standard deviation) resulting in 36 (instead of 37) dimensions for `MFCC+RQA-900-w.o.C0` and 25 (instead of 26) for `MFCC-900-w.o.C0`.

Except for the case where no gain is applied (at 0dB), the systems including RQA features are affected by variations in the energy level. This is probably because the similarity matrix between consecutive MFCC frames (used to extract RQA features) is highly dependent on C0. On the contrary, `MFCC-900-w.o.C0` system seems to be more robust to energy variation.

Many prior works identify the energy as a discriminant factor to separate and distinguish different contexts [27, 38, 53, 54, 59]. In fact, all these methods computed features which represent the amount of energy of several sub-bands with respect to total energy. In this way the system will depend upon a relative measure of energy, in constrast to an absolute measure (e. g. C0) which is less robust to changing conditions.

3.4.2 *The frequency range*

The next set of experiments investigate the frequency range from which MFCC features are extracted: `MFCC+RQA-900` computes MFCCs over a $[0, 900]$Hz range [56]. This choice may be optimal for the DCASE 2013 database, but may not generalise well to other datasets. The hypothesis is that the type and nature of acoustic scenes determines the optimal range of frequencies.

The first experiment shows on Fig. 13 (b) accuracy as a function of frequency range. The lower bound is fixed at 0Hz while the upper frequency varies from 900Hz, to 8000Hz passing through 2000Hz and 4000Hz. Note that the systems are re-trained for each frequency range and then tested with the same conditions.

`MFCC+RQA` and `MFCC+RQA-w.o.C0` systems achieve their best accuracy for the 0-900Hz range and their worst at $[0, 8000]$Hz. The other two systems `MFCC+RQA` and `MFCC+RQA-w.o.C0`

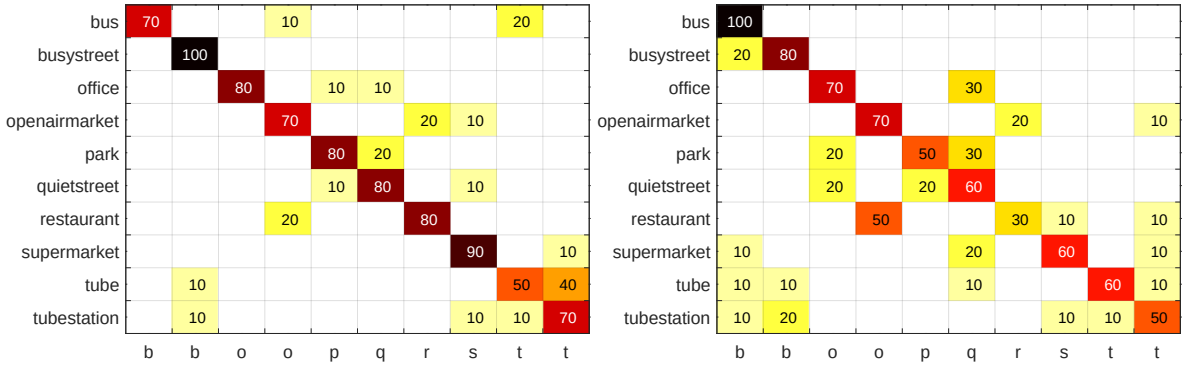


Figure 14: The confusion matrices of (a) configuration `MFCC+RQA-900` and (b) `MFCC+RQA-8000` on the right. Each confusion matrix expresses the actual label on the rows and the predictions on the columns. The value in each block are computed as the number of correctly classified samples of a class divided by the total number of samples of the same class.

(without the 11 RQA features) reach 70% accuracy in the range $[0, 2000]$Hz. All systems exhibit their worst performance at 8000Hz.

The second experiment illustrates the effect of frequency ranges for two systems: `MFCC+RQA-900` adopts a $[0, 900]$Hz range; `MFCC+RQA-8000` uses a $[0, 8000]$Hz range. The two confusion matrices are displayed side-by-side in Fig. 14: `MFCC+RQA-900` to the left and `MFCC+RQA-8000` to the right.

Interestingly, performance for an acoustic scene subset is affected significantly by the frequency range. In particular, the performance for *park*, *restaurant*, *supermarket* and *tubestation* scenes deteriorate as the frequency range increases/decreases. In constrast, performance for *bus* scene improves from 70% to 100%.

These results give some insights into how MFCCs encode spectral information depending on the number of mel filters. A higher resolution in the lower frequencies generally helps to discriminate between scenes in the DCASE 2013 database. The same reasoning may be extended to other scenes which contain discriminative information at higher frequencies.

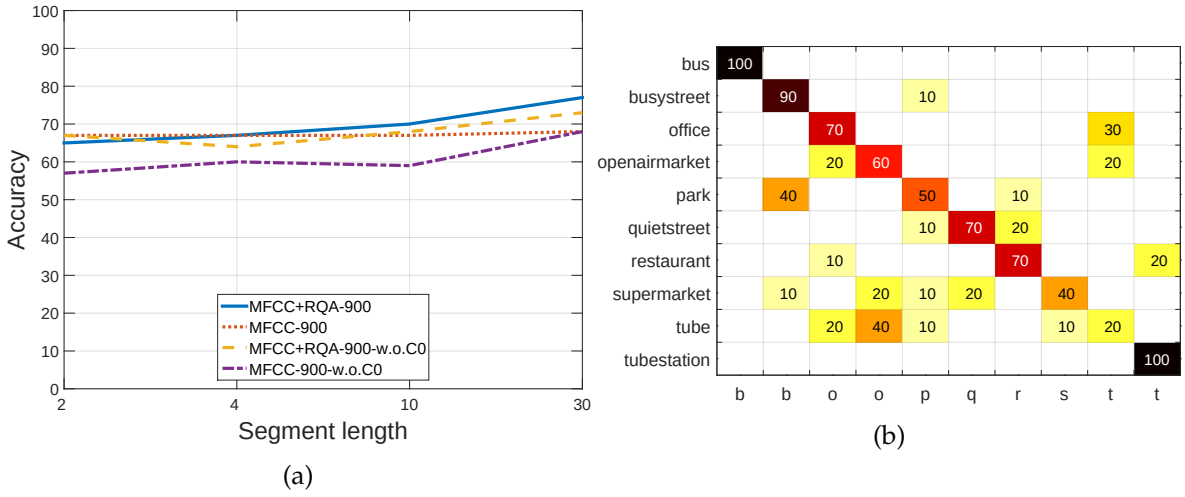
To conclude, MFCCs encode the information of the spectrum with respect to the frequency range. The same number of mel-filters can be applied to different frequency ranges thereby producing a higher resolution in different parts of the spectrum. This aspect influences the classification of certain scenes to the detriment of others.

3.4.3 *Integration of segments over time*

The `MFCC+RQA-900` baseline system averages both frame-level MFCCs and RQA to get a single feature vector for 30s. A different approach is tested in this section: instead of computing features over 30s segments, shorter segment durations are considered.

In practice, mean and standard deviation of MFCCs and the mean of RQA features are calculated from different segment lengths. The lengths considered are $[2, 4, 10, 30]$ seconds and the overlap is always 50% of the segment duration. A SVM classifier is then trained and tested using different segment lengths. In the case of segments shorter than the file duration (30s for DCASE 2013 recordings), a majority vote scheme is employed to obtain a single decision.

Recognition accuracy is illustrated in Fig. 15 (a) as a function of the segment lengths. Also illustrated Fig. 15 (b) is the confusion matrix for the `MFCC+RQA-900` system with a segment length of 4s and 50% overlap. As expected all systems reach their peak at 30s, when MFCCs and RQA features are computed over 30 seconds of signal. Statistics calculated over smaller



(a) (b)

Figure 15: (a) the accuracy as a function of different segment lengths. The features depicted in the legend are computed through statistics (mean and standard deviation). For segments smaller than the audio file duration, a majority vote scheme is employed to obtain a single decision for each audio file; (b) the confusion matrix of the system `MFCC+RQA-900` computed at 4 seconds.

segments which are then integrated over time do not increase accuracy. We also observe that the `MFCC-900` system maintains stable behaviour from 2 to 30 seconds.

In general a poor resolution in the frequency domain has a greater impact on performance than a poor temporal resolution: the accuracy of `MFCC+RQA-8000` system at 30s is 6% less than the accuracy of the `MFCC+RQA-900` system at 4 seconds. At the same time, some classes (e. g. *park*, *supermarket*, *tube*) require features computed over longer segments. On the contrary, *bus* and *tube* classes seem to be recognised more reliably at 4 seconds.

Similar to frequency range, the duration of the segment used for feature extraction influences recognition accuracy significantly: acoustic characteristics of a scene can be more favourable to one segment length rather than another. If one had, for instance, to separate only *bus* or *tube* classes, it would be preferable to choose the 4 seconds configuration. Hence, also segment duration can be adapted to the number and type of acoustic scenes.

3.4.4 *Impact of temporal derivatives*

The problem of representing the temporal evolution of an acoustic scene is still an open issue. State-of-the-art ASC approaches based on MFCC statistics average temporal information of frame-level features. Other classifiers (e. g. GMM) model the MFCC distribution without modelling the temporal evolution. Other techniques which model the temporal structure (such as HMM) are rarely used for ASC [27, 46]. The only HMM system submitted to DCASE 2013 used features computed over both short and long frames (25ms and 1.5s) [59]. HMMs assume that each frame-level feature is independent.

Different to speech or music analysis, the temporal evolution of an acoustic scene lacks an ordered structure: prominent sounds may appear in any order, making it difficult to model their temporal evolution. On the contrary, dynamic features complement static MFCCs with a tunable degree of temporal information with no need for explicit modelling.

For all frame-level features (e. g. MFCCs), time derivatives for consecutive frames can be extracted. These experiments, reported below, show a comparative study of first and second order temporal derivatives. First and second order derivatives Δ and $\Delta\Delta$ incorporate the temporal evolution creating a dynamic feature vector of MFCCs [74] centred in (t) with a

System	static	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 7$
MFCC-Δ,$\Delta\Delta$-900_SVM	68% ($\pm$7%)	62% ($\pm$12%)	67% ($\pm$12%)	61% ($\pm$8%)	55% ($\pm$12%)
MFCC-Δ,$\Delta\Delta$-900_GMM	55% ($\pm$4%)	69% ($\pm$9%)	69% ($\pm$5%)	69% ($\pm$9%)	65% ($\pm$1%)

Table 3: The effect of the window τ used to calculate the Δ and $\Delta\Delta$ features. Each MFCC coefficient is centred in the middle of the window in order to have the same number of temporal coefficients τ before and after the coefficient $x_i^{(t)}$ at time t. The table reports results from DCASE 2013 evaluation set.

temporal coefficient of τ which determines how many frames should be considered before and after (t):

- 13 coefficients MFCC $x_i^{(t)}$ at time t
- 13 first order derivative coefficients $\Delta x_i^{(t)} = x_i^{(t+\tau)} - x_i^{(t-\tau)}$
- 13 second order derivative coefficients computed from $\Delta\Delta x_i^{(t)} = \Delta x_i^{(t+\tau)} - \Delta x_i^{(t-\tau)}$

The final feature vector $\boldsymbol{x}_n$ is then a combination of static and dynamic features computed over D dimensions $\boldsymbol{x}_n = [x_i, \Delta x_i, \Delta\Delta x_i]_{i=1\dots D}$. Depending on the type of classifier, further processing may be required:

- for the SVM classifier, the feature vector is obtained by averaging the Δ and $\Delta\Delta$ over 30 seconds. Averaged temporal derivatives are then added to MFCCs statistics. The final dimension of $\boldsymbol{x}_n$ is then 52. This system is referred to as MFCC-Δ,$\Delta\Delta$-900_SVM;
- for the GMM classifier, time derivatives are concatenated to static MFCCs at the frame-level, thereby being a 39-dimensional feature vector. This system is referred to as MFCC-Δ,$\Delta\Delta$-900_GMM.

The following set of experiments focuses on the window length used in the time derivative. Results in Tab. 3 present accuracy and confidence intervals (CIs) for different values of τ.

The best configuration for the MFCC-Δ,$\Delta\Delta$-900_SVM system achieves an accuracy of 67%, with $\tau = 3$ MFCCs frames (3 before and 3 after the current (t) MFCC). This performance is equivalent to that of static-only system MFCC-900 and 10% worse than that of the MFCC+RQA-900 system. These experimental results suggest that the extraction of features from 30 seconds segments destroys useful information that is therefore captured in time derivatives. In addition, the amount of data needed to support a higher dimensional feature space often grows exponentially with dimensionality. In this case, passing from 26 to 52 feature dimensions would require more data to reliably model those features. This phenomenon is referred to in the machine learning literature as the *curse- of-dimensionality* [75].

The second system MFCC-Δ,$\Delta\Delta$-900_GMM shows the benefit of temporal derivatives. Performance is set for a temporal coefficient τ of 3. This configuration achieves an accuracy of 69% with a lower CI interval. GMMs model the temporal derivatives better than SVMs improving upon the performance of the baseline system by 14%.

Experiments suggest the importance of temporal information for recognising an acoustic scene. On the one side, RQA features represent this information with a compact representation of the similarity between 40 consecutive MFCCs; on the other side, time derivatives Δ, $\Delta\Delta$ combined with static MFCCs increase the accuracy of the GMM baseline system.

3.5 CONCLUSIONS

This chapter analysed in detail the winning system of the DCASE 2013 ASC evaluation, starting with the re-implementation and assessment of the same system reported in the literature. The SVM classifier was presented in Sec. 3.2 with a particular focus on generalisation through a grid-search strategy. All modifications made to feature extraction block (see Fig. 3) have shown a significant impact on performance, showing how feature design can influence classification of acoustic scenes. The main limitations of this baseline system are therefore related to the type of features, composed of MFCCs and RQA features.

As an example, MFCCs have been designed for speech recognition, but do not optimally describe the complexity of an acoustic scene (i. e. overlapping sounds or additive noise could change part of the spectrum and, since MFCCs encode the spectrum as a whole, the resulting features will be affected by such changes). In addition, RQA features do not seem to be robust to energy, frequency-range or segment-length variations. Beside these considerations, temporal information is not realiably and usefully captured due to the sparse temporal structure of acoustic scenes.

The effectiveness of the current system is stricly related to the set of scenes in the DCASE 2013 database. The experimental works presented in this chapter suggest that the same system may poorly generalise on other datasets, characterised by a different composition of classes, recording conditions and file duration. Thus, the feature adaptability to different databases will be used as a guideline in further ASC evaluation.

To conclude, features have a significant influence on ASC performance and their impact seems to be linked to the composition of the scenes to be recognised. The analysis of the feature-scene relation and the evaluation over different databases will be the content of the next chapter.

4

VISUALISING AND ANALYSING FEATURES

Experiments presented in Chapter 3 demonstrate that modifications in feature extraction and post-processing have a significant impact on performance. This finding can be extended to a more generic hypothesis: there exists no unique feature set capable of discriminating all possible scenes. In fact, features should be optimized according to the set of scenes one seeks to classify.

Works in this chapter relate to the visualisation and the analysis of features with respect to the scenes they represent. The visualisation of high-dimensional features with a low-dimensional representation may help to discover hidden relationships between classes or between samples of the same class. Just looking at the final classifier performance does not provide enough insights about the feature distribution and its relationship to the acoustic scenes. In addition, each visualised sample directly relates to a single audio sample. Visualising the sample distribution and listening to corresponding audio excerpt gives a deeper interpretation of high-dimensional features.

Visualisation and listening not only help to better understand the data, but also helps to improve the feature design. Visualisation provides *qualitative* analysis as a generic indication of separability. Nevertheless, more reliable metrics should complement the information provided by the visualiser. These metrics *quantify* the feature discrimination power. Both *qualitative* and *quantitative* evaluations were used in the thesis works as an *exploratory* tool to validate the feature representation of the acoustic scenes. The features proposed in this chapter take advantage of the findings coming from the visualisation and feature analysis.

The remainder of the chapter is organised as follows: detailed explanations about the visualisation technique is provided in Sec. 4.1; feature metrics are described in Sec. 4.2; the collection of a larger dataset is in Sec. 4.3; the application of feature analysis to feature design is reported in Sec. 4.4; conclusions in Sec. 4.5 discuss the need for visualisation and feature metrics for a deeper understanding of the ASC problem.

4.1 VISUALISING HIGH-DIMENSIONAL FEATURES

The problem of visualising high-dimensional features has gained importance in data mining community [76]. According to the main definition, data mining extracts meaningful information from data by using several techniques such as machine learning, statistical analysis, database indexing and data processing. Data visualisation is part of data mining realm while providing complementary information to standard machine learning techniques.

Visualisation allows to discover connections and similarities between classes, giving an interpretation to multi-dimensional features. The hypothesis of the work reported here is that a single set of features is not optimal for the classification of all possible classes. Hence, a good visualisation can help in understanding the intrinsic structure of the data and in designing discriminant features.

There exist several visualisation techniques which embed high-dimensional spaces into a 2 or 3-dimensional space while maintaining the arrangement or structure of the original data.

The common idea behind many of these techniques is to find a mapping function where the distance in lower-dimension space reflects similarities in the original high-dimensional space. This is, for example, the case of PCA which finds a linear projection of the original data in such a way that the variance of the projected data is maximised [62]. Nevertheless PCA is sub-optimal for modelling data which are non-linearly distributed. Real, complex data (such as acoustic scenes) may contain local and global structures which cannot both be captured by standard dimensionality reduction techniques.

To solve this problem, a technique called t-distributed stochastic neighbor embedding (t-SNE) [77] has been adopted as an ASC feature visualiser. The goal is to find a non-linear transformation such that a set of samples in high-dimensional space will be represented meaningfully in a lower-dimensional space, typically a 2-D plane. t-SNE is non-linear because it applies local transformations to different regions of the feature space. Mathematical details of this visualisation method are reported in Annex A.2.

Even though t-SNE provides good visualisation insights, interpretation of results is subject to debate [78]. First, the perplexity may affect the quality of the mapping between high-low dimension transformation. A second aspect relates more to a generic interpretation: cluster sizes should not be evaluated by their variance because t-SNE adapts to different densities. The interpretation concerns more the similarities or local relationships between samples of different classes. Herein are listed the main criticisms:

- the *perplexity* (i. e. the estimation of the number of neighbours close to each point) is a global parameter and it affects the final embeddings. When the perplexity is too low with respect to the number of samples per class, only the local characteristics will be retained. On the contrary, when the perplexity is too close to the number of samples per class, the algorithm will have unpredictable results. Empirically, the perplexity value should be smaller than the number of samples per class, but big enough to capture the global structure of the data;
- successive runs don't produce that same results due to the non-convex cost function optimised by gradient descent;
- t-SNE can be accelerated using a tree-based algorithm called *Barnes-Hut*, making possible the application of t-SNE to millions of samples [79]. The trade-off parameter, θ, between estimation error and fast approximation has to be selected before running the algorithm. A value of θ close to 1 reduces computational complexity at the expense of greater errors; a value close to 0 is more computably demanding, but results can be more meaningful.

4.1.1 *t-SNE for ASC*

Presented here is the application of t-SNE to the ASC problem, using the DCASE 2013 evaluation dataset. The goal is the analysis of three systems using both t-SNE and classification confusion matrices. These systems are listed below:

1. `MFCC+RQA-900`(RNH) represents the state-of-the-art system as reported in Chapter 3. This systems is characterised by MFCCs and recurrence quantification analysis (RQA) features computed in the range $[0, 900]$Hz. Average and standard deviation are extracted over interval of 30s. The number of samples for each class is 10;
2. `MFCC-2000-4s` is the second best system in term of performance among the systems tested in Chapter 3. This system represents a first attempt to solve limitations such as energy, segment length and frequency range. The choice of these parameters comes

directly from results in Figs. 13 (a), 13 (b) and 15. MFCCs with the first coefficient C0 and a range of $[0, 2]$kHz are computed with a frame size of 32ms overlapped by 16ms. Statistics in the form of average and standard deviation are then extracted over segment of 4s overlapped by 2s. The number of segments for each 30s of audio signal is 14, resulting in 140 samples for each acoustic scene.

3. `MFCC-2000-4s-.w.o.C0` refers to the aforementioned system without C0.

In all experiments, the first step is to find the principal component coefficients which retains 98% of the variation present in the original data. This removes redundant dimensions and improves the computational efficiency of t-SNE, which is then applied to the PCA projected space. t-SNE returns a 2D map which is then shown as a scatter-plot. In that sense, t-SNE is completely unsupervised and reflects similarities between samples with no class information. Class labels are not used to determine spatial coordinates. Colours/markers are added subsequently. Due to the modest size of the DCASE 2013 dataset, the trade-off θ between speed and estimation error is empirically set to 0.7.

Interestingly, t-SNE maps in Fig. 16 exhibits a degree of correlation with the corresponding confusion matrix obtained after SVM classification. t-SNE uses features computed at the segment-level (e. g. 4s overlapped by 2s). To compare visualizations at segment-level to the class misclassification, corresponding confusion matrices are computed from segment-level predictions. The group of overlapping samples in the lower right corner of Fig. 16 (a) represents acoustic scenes which are acoustically similar such as *office*, *quietstreet* and *park*. Misclassification of three 'similar' scenes is observed in the corresponding confusion matrix of Fig. 17 (a).

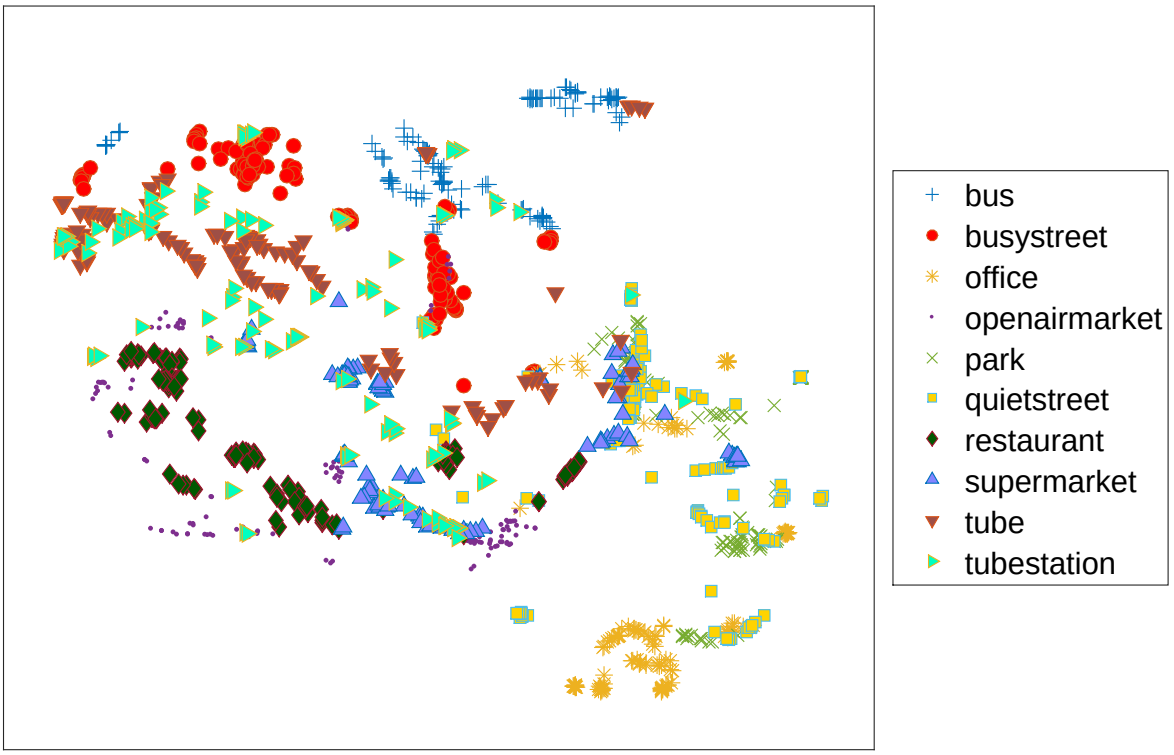
The *openairmarket* class is overlapped with *restaurant* and *supermarket*. The same behaviour is observed between *supermarket* and *quietstreet/tubestation*. Also in this case, the same errors can be found in the confusion matrix (Fig. 16 (a)). The last observation concerns the *tubestation* scene which is literally spread across the entire feature space. In general, distances in t-SNE embeddings reflect intuitive acoustic similarities: *bus*, *busystreet* and *tube* form a *transport mode* cluster; *restaurant*, *openairmarket* and *supermarket* form another group; finally *office*, *quietstreet* and *park* form a third group of acoustically-related scenes. Note that, in this first experiment, MFCC features include C0 which expresses the energy level of a scene. C0 may be an important factor in grouping these classes together.

The same systems are reported here without C0 (Figs. 16 (b) and 17 (b)). In this experiment, the general distribution of the classes is less clustered with more overlapping between samples of different classes. `MFCC+RQA-900` system extracts a single feature vector from the entire 30s audio signal. Since there are fewer samples per class, perplexity has to be adjusted to the number of samples per class. The most compact representation is empirically chosen with perplexity equal to 5. In the t-SNE visualisation of `MFCC+RQA-900` system (Fig. 18) samples of the same class seem clustered while being well separated from samples of other classes. The corresponding confusion matrix is reported in Fig. 14 (a).

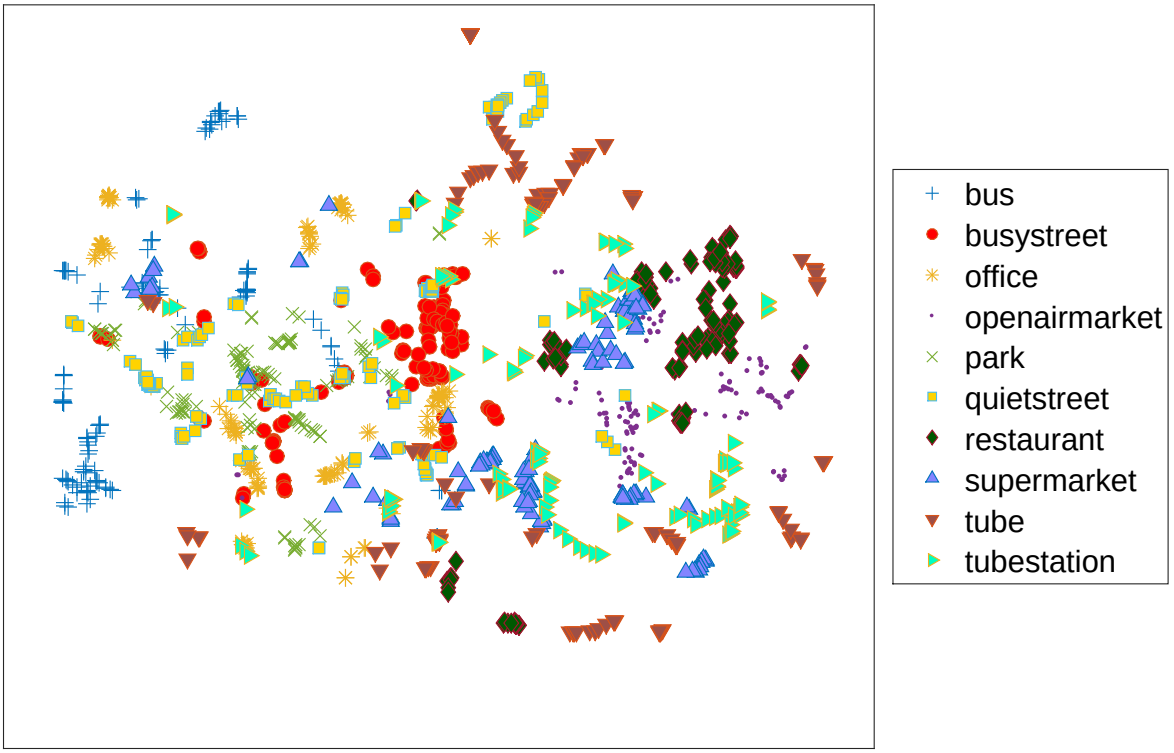
4.1.2 *Insights of visualisation*

Identification of patterns in feature space is essential for better understanding of the ASC problem. Visualisation allows researchers to easily identify these patterns and to inspect each sample. Albeit being an unsupervised method, t-SNE provides a representation which complement that of the classification.

Observations derived from t-SNE application to the DCASE 2013 dataset are reported in the following. A first observation concerns the level of spread of some class samples in experiments with 4s segments. Acoustic scenes represented with t-SNE show a multi-modal

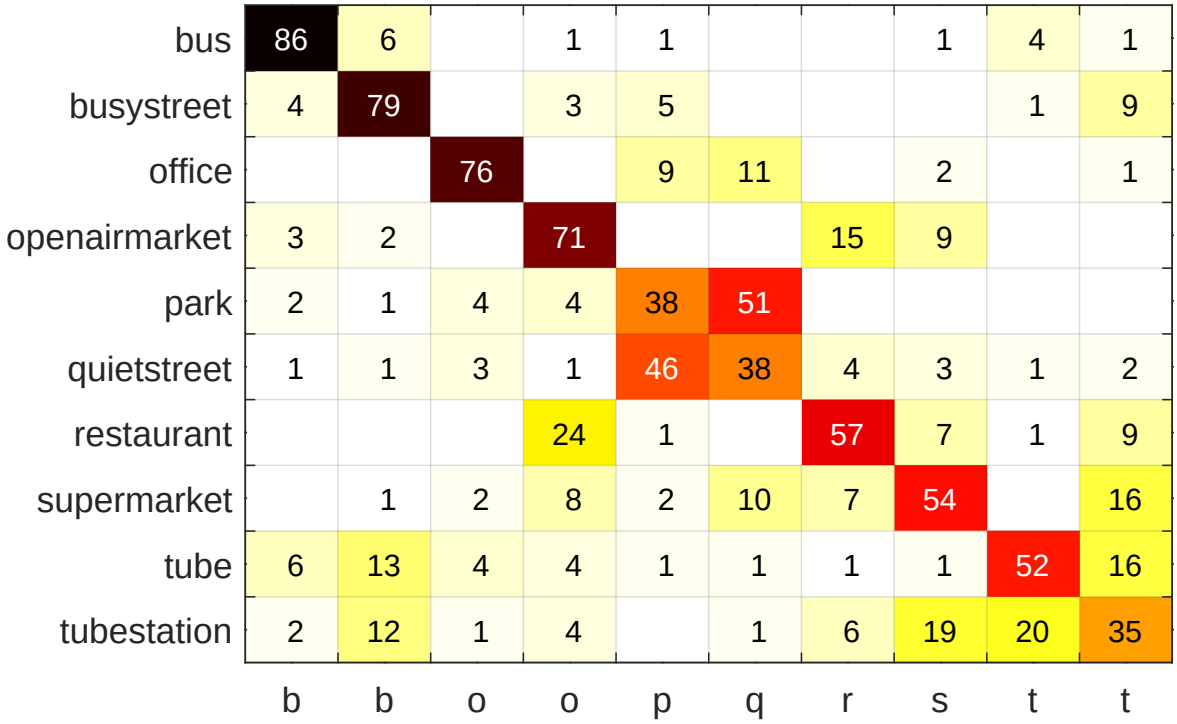


(a)

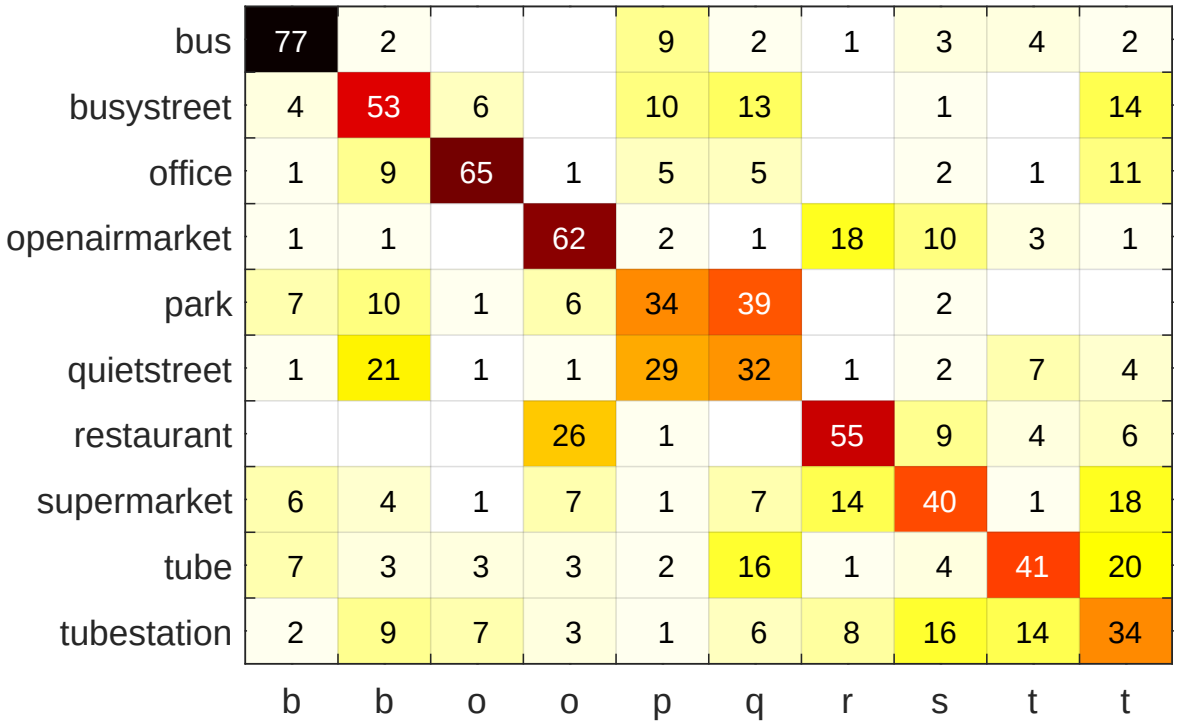


(b)

Figure 16: t-SNE visualisation with perplexity equal to 50 for (a) `MFCC-2000-4s` and (b) `MFCC-2000-4s-.w.o.C0`.



(a)



(b)

Figure 17: Corresponding confusion matrices of (a) `MFCC-2000-4s` and (b) `MFCC-2000-4s-w.o.C0`. Confusion matrices are computed over segment-wise predictions, where no majority vote strategy is used.

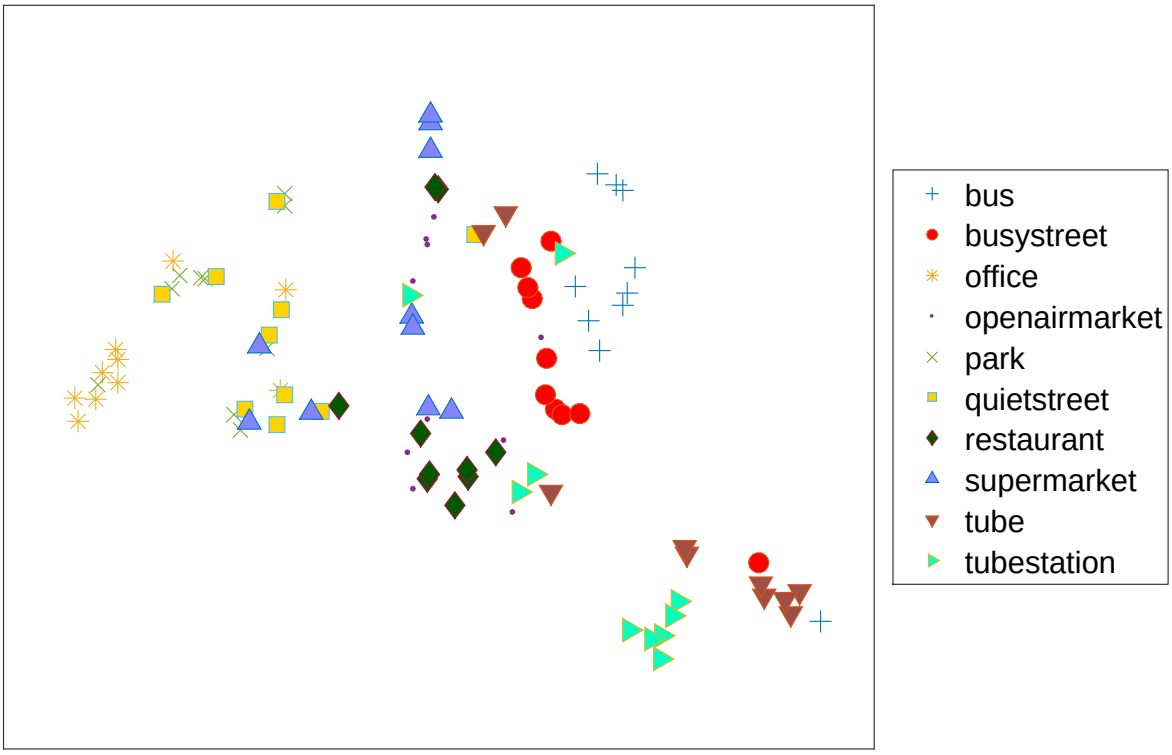


Figure 18: t-SNE embeddings on 2D plane for `MFCC+RQA-900Hz`. Perplexity is set to 5.

distribution: samples from the same class lie in distinct clusters. Upon inspection of samples in each cluster, it is observed that each cluster represents samples from the same recording. As an example, let us consider the *busystreet* samples identified by red circles (Fig. 16 (a)). Samples closely located in the t-SNE visualisation represent segments coming from the same recording. Almost all scenes exhibit this multi-modal behaviour.

A second observation regards the impact of C0 on the t-SNE visualisation. It is visible from the embeddings of Fig. 16 how C0 leads to a better separability. In Fig. 16 (b), samples of different classes are overlapped and lose the structure seen in MFCC-2000-4s system. On one side, the importance of energy is one more time highlighted; on the other, it can be derived that the modest size of the DCASE 2013 database makes energy determinant for the classes discrimination.

Eventually, depending on the type of classes, some features may give better separation than others. t-SNE visualisation provides some insights into relationships between samples and classes. As an example, *park*, *office* and *quietstreet* classes are grouped together and may be difficult to distinguish. *Bus*, *busystreet*, *tube* and *tubestation* share acoustic properties and form a distinct group. Similar behaviour is seen for *restaurant*, *park* and *openairmarket*.

t-SNE mappings complement classifier accuracies and confirm misclassifications observed in the confusion matrix. t-SNE was thus adopted as a data exploration tool for ASC.

4.2 FEATURE METRICS

Results show consistent relationships between t-SNE mappings and confusion matrices. Although being a powerful tool for data exploration, t-SNE cannot provide reliable and quantitative metrics. This is due to the gradient descent used during optimisation and to the global parameters (e. g. perplexity) which alter reproducibility of results.

In order to understand more about a high-dimensional feature space, *quantitative* metrics are required to reflect the separability of a set (or a subset) of feature. Feature metrics have gained importance in many research fields [80]. There are several benefits to feature analysis:

reducing training storage and computation, improving data understanding and augmenting classifier performance. In ASC, feature metrics provide a complementary information to that coming from t-SNE.

According to [81], techniques for measuring feature separability can be subdivided in three main groups: *wrappers* methods use classifier scores to rank the features; *embedded* methods implement the feature selection method in the classifier optimization function; *filter* methods analyse intrinsic properties of the data independently of the classifier.

In order to be completely independent from the classifier, *filter* methods are preferred herein. *Filter* methods rank features according to some metrics which represent the discriminative power of each feature taken alone. Although being computationally efficient, these methods do not capture inter-correlation between features. Thus, it may happen that a feature with lower rank would be combined usefully with another. Since the focus of this chapter is more on data understanding, the problem of feature combination has not been taken into consideration. Hence, the well known Fisher score [82] has been adopted to select features and to evaluate a global separability metric. More details will be provided in Sec. 4.2.1.

To highlight the dependency between features and acoustic scenes, a metric based on Bhattacharyya distance is also discussed in Sec. 4.2.2. This metric measures the amount of overlap between a pair of two class distributions and is therefore capable of ranking pairs of acoustic scenes by their discriminating capacity [83].

4.2.1 *Fisher score*

The key idea of the Fisher score is to measure the ratio between samples of different classes and samples in the same class [84] Ideally, samples of one class close should be located together in feature space and, at the same time, they should be well separated from samples of other classes. These two measures are identified respectively by the inter-class $\boldsymbol{S}_b$ and intra-class $\boldsymbol{S}_m$ variance matrices:

$$\begin{aligned} \boldsymbol{S}_b &= \sum_{c=1}^{C} N_c(\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^T \\ \boldsymbol{S}_w &= \sum_{c=1}^{C} \sum_{\boldsymbol{x}_n \in \mathcal{X}_c} (\boldsymbol{x}_n - \boldsymbol{\mu}_c)(\boldsymbol{x}_n - \boldsymbol{\mu}_c)^T, \end{aligned} \tag{17}$$

where for C classes, $\boldsymbol{\mu}$ represents the mean of all samples, $\boldsymbol{\mu}_c$ is the mean of the samples of the c^{th} class, $\mathcal{X}_c$ represents the set of training samples $\boldsymbol{x}_n$ belonging to the c^{th} class and N_c is the cardinality.

Ideally, $\boldsymbol{S}_b$ is higher whereas $\boldsymbol{S}_m$ is low, thereby maximising the Fisher criterion F

$$F = \frac{\mathrm{tr}(\boldsymbol{S}_b)}{\mathrm{tr}(\boldsymbol{S}_w)}. \tag{18}$$

The objective of function F is to group all samples of each class close to their mean and to ensure that class clusters are well separated. Together with the criterion F which expresses the separability of all features as a whole, a dimension-dependent Fisher score F_i can be computed for each dimension of features in $\mathcal{X}$. Heuristically, each dimension is evaluated independently instead of computing all possible combinations [80]. With this method, features that may have a higher score if combined together are not considered.

Let us define $\mu_{c,i}$ and $\sigma_{c,i}$ as the the mean and standard deviation of the i^{th} feature dimension of the c^{th} class. With μ_i and σ_i we indicate the mean and standard deviation

System	J	Accuracy
MFCC+RQA-900	2.73	77%
MFCC-2000-4s	1.84	71%
MFCC+RQA-900-w.o.C0	0.95	70%
MFCC-2000-4s-w.o.C0	0.46	61%

Table 4: Fisher score F for different ASC systems. Aside are reported accuracies from SVM classifier.

of the i^{th} feature over the whole dataset. The Fisher score for the i^{th} feature dimension is then given by:

$$F_i = \frac{\sum_{c=1}^{C} n_c(\mu_{c,i} - \mu_i)^2}{\sigma_i^2} \tag{19}$$

where $\sigma_i^2 = \sum_{c=1}^{C} n_c \sigma_{c,i}^2$. Results for global Fisher score F are reported in Tab. 4 for systems presented in this chapter, with the addition of the state-of-the-art system without the C0 (MFCC+RQA-900-w.o.C0). The systems are : MFCC+RQA-900, MFCC+RQA-900-w.o.C0, MFCC-2000-4s and MFCC-2000-4s-w.o.C0. The Fisher score is computed over all samples of the DCASE 2013 evaluation set, in order to be comparable with t-SNE plots and confusion matrices presented earlier.

In addition to the global Fisher score F, a ranking of each dimension is also detailed in Fig. 19. Results illustrated in Tab. 4 show that the DCASE 2013 state-of-art system MFCC+RQA-900 has the higher separability with a Fisher score of 2.73. Interestingly, a drop in the Fisher scores corresponds to a drop of accuracy.

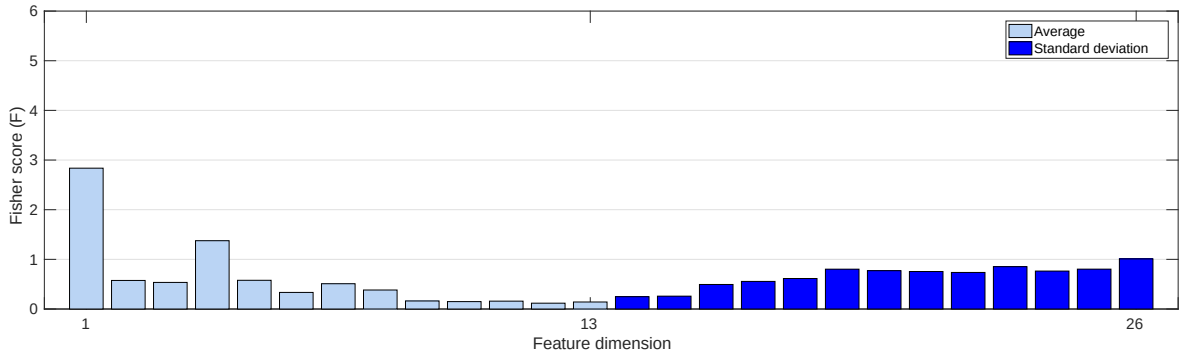
Without the first MFCC coefficient C0, all systems provide lower scores with respect to their C0 versions. As an example, the impact of removing C0 is evident for MFCC-2000-4s and MFCC-2000-4s-w.o.C0 whose Fisher score passes from 1.84 to 0.46. This difference corresponds to a drop in accuracy of 10%. As previously mentioned, the impact of removing C0 (the only feature depending on the energy level in the signal) is high, suggesting that the scenes in DCASE 2013 may be distinguished just by their energy level.

A similar conclusion comes from results for each feature, in Fig. 19. Reported here the Fisher score as function of each dimension F_i for two systems MFCC-2000-4s and MFCC+RQA-900. In the top figure, the Fisher score for C0 is three times greater than the remaining feature scores. In the bottom figure, RQA features (in red bars) contain the highest Fisher score, even though the inclusion of C0 reports one of the highest Fisher scores.

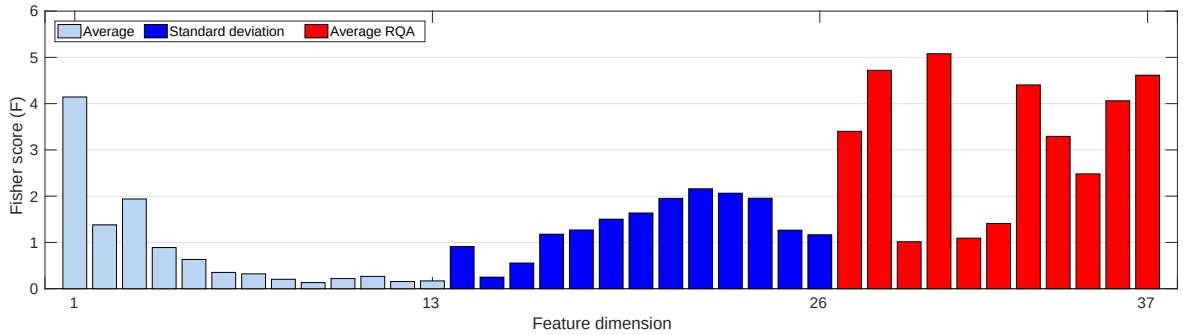
4.2.2 *Bhattacharyya distance*

That some acoustic scenes are easier to separate than others has been shown already through t-SNE visualisations. Even though t-SNE visualisations provide a qualitative evaluation, a quantitative distance is required to measure the separability between pairs of classes. This distance will produces a *ranking* of class pairs, from the easiest to the most difficult to distinguish. The distance-based ranking: i) quantifies the separability between acoustic scenes; ii) shows which class pairs are the most difficult to distinguish.

Among other techniques [85], the *Bhattacharyya* distance provides this separability measure [86]. The main idea is to extract a distance from parametric distributions. Supposing that



(a)



(b)

Figure 19: Fisher score (F_i) as a function of different feature dimensions composing the feature vector. In (a) the system `MFCC-2000-4s` is presented in the bar plot. Average and standard deviation are computed every 4s overlapped by 2, resulting in a 26 dimensional feature vector. This split between average and standard deviation is highlighted with the light blue for the former and dark blue for the latter. In (b) the system `MFCC+RQA-900` presents the same procedure for the average and standard deviation to which RQA features are appended. Fisher scores of these features are respectively in light blu, dark blu and red.

two acoustic scenes respect normal distributions, the Bhattacharyya distance D_B between classes c_1 and c_2 is given by:

$$D_B(c_1, c_2) = \frac{1}{8}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \left(\frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{2}\right)^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2}\log \frac{\det\left(\frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{2}\right)}{\sqrt{\det(\boldsymbol{\Sigma}_1)\det(\boldsymbol{\Sigma}_2)}}, \quad (20)$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ denote the mean and covariance matrix of the two classes. The Bhattacharyya distance in Eq. 20 includes two terms which both reflect the separability measure: the first term expresses this separation through the difference between the class means; the second term relates to class covariances. Note that, when the two means are equal, B_D depends only on the covariance term; when the two covariances are equal, B_D depends on the means alone.

Another aspect concerns the determinants of the covariance matrices: when the feature dimensionality D is greater than the number of samples per class n_c, the $\text{rank}(\boldsymbol{\Sigma}_c) < D$ and therefore $\det(\boldsymbol{\Sigma}_c) = 0$. The consequence of this is that systems for which $n_c < D$ cannot be analysed with the Bhattacharyya distance. Thus, systems for which Bhattacharyya distance is applicable are `MFCC-2000-4s` and `MFCC-2000-4s-w.o.C0`. These systems extract features over shorter segments thereby producing a number of samples per class which is greater than the feature dimensionality D.

The objective of the following experiments is to measure the separability between pairs of acoustic scenes with the Bhattacharyya distance. This should reflect the t-SNE visualisations while providing a quantitative measures. The Bhattacharyya distance as a function of all possible class pairs is depicted in Fig. 20: distance for the `MFCC-2000-4s` system are displayed by a solid black line while those for the `MFCC-2000-4s-w.o.C0` system are shown by a black dashed line. Class pairs are displayed on the x axis, ordered according to the ranking distances obtained for the `MFCC-2000-4s` system. For the sake of clarity, distances B_D for the `MFCC-2000-4s-w.o.C0` system are displayed in the same plot but respecting the class ranking of the first system. In this way, differences in term of B_D between the two systems are clearly visible.

Upon first sight, the system with C0 has a greater separability power. More specifically, the class pairs corresponding to the highest values are *bus-office*, *busystreet-office*, *busystreet-restaurant*. The second system follows a similar trend with some differences: *bus-office*, *busystreet-office* and *office-tube* face a significant drop in terms of Bhattacharyya distance. Classes with the lowest B_D values are respectively *openairmarket-restaurant*, *park-quietstreet* and *tubestation-tube*. Interestingly, these findings confirm a lower separation in t-SNE visualisation (Fig. 16 (a)). The same classes report a higher rate of misclassification in the confusion matrix (Fig. 17 (a)).

4.2.3 *Insights of feature metrics*

Findings coming from feature analysis complement and confirm those of t-SNE visualisation. Both Fisher scores and Bhattacharyya distances validate the influence of the first MFCC coefficient C0 on the results. As demonstrated by Fisher scores (Tab. 4) and by Bhattacharyya distances (Fig. 20), the performance of all systems is affected by the exclusion of C0. For the DCASE 2013 database, most of the separability power relates to the energy level (represented by C0) rather than the other features.

Results coming from Bhattacharyya ranking confirm the initial hypothesis of the scene-dependent features. In fact, if the dataset was composed of the most separable classes (according to results in Fig. 20, *bus, office, busystreet, restaurant*), the current MFCC-based

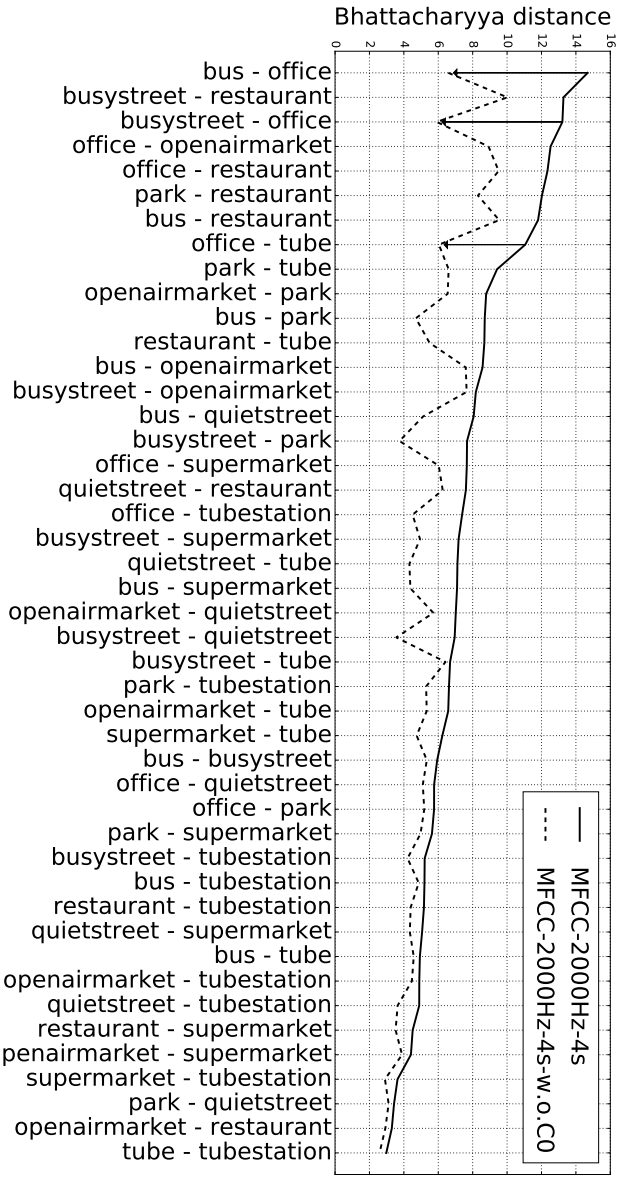


Figure 20: Bhattacharyya distance B_D for each pair of acoustic scenes of DCASE 2013 evaluation set. In solid black line the `MFCC-2000-4s` results are presented; in dashed black line the `MFCC-2000-4s-w.o.C0`. The ranking of paired classes on the x-axis is ordered according to the `MFCC-2000-4s` distances. The arrows indicate the classes which have the highest difference in term of B_D between the two systems.

Context	Total time	meta-tag options
Bus	9h8m	Position, road-type, occupancy, windows
Car	3h40m	Type, position, road, occupancy, windows
Office	3h26m	Occupancy
Subway	10h30m	Occupancy
Street	2h42m	Location

Table 5: Duration of recordings for each context in the NXP database beside of associated meta-tag options.

features would be sufficient to separate them. These learnings justify the works of the following sections and later in the thesis:

- the DCASE 2013 dataset is not suited for exhaustive evaluations. Due to its modest size, the exclusion/inclusion of a single feature (C0) is significant enough to change the classes distribution and to impact the performance by 10%. Hence, a dataset containing about 30 hours of recordings was collected. This dataset is referred to as NXP dataset and is described in sec. 4.3. It is also argued that cross-dataset evaluation provides more information about the capacity of a method to adapt to different conditions (e. g. number and type of classes, recording conditions, recording lengths) thereby having a more realistic view on the ASC problem;
- all systems accuracies are impacted by removing C0 from the feature vector. At the same time, systems which extract C0 are shown to be less robust to energy scene variations (Chapter 3). This proves that an indication of energy level between acoustic scenes augments the separability power and should be taken into consideration when designing ASC features. The proposed features, which replace C0 with a *relative* measure of energy, are discussed in sec. 4.4.

4.3 NXP DATASET

The variety, quality and consistency of audio recordings are key factors in designing realistic ASC systems. The NXP dataset was recorded by volunteers using different vendors mobile devices where a recording application was installed. The application handles both data collection and labelling. The sampling frequency is set to 16kHz (in contrast to the 44100Hz of the DCASE 2013 database). The original recordings with the corresponding labels are then uploaded to a centralized server. Labels and associated meta-tags (e. g. description of the scene) are selected among a close list of possible names. The main families of meta-tags are listed in Tab. 5 and defined as: position (front, middle, back), occupancy (crowded, normal, empty), windows (open, close), type (petrol, electric, sport), road (highway, city, country road) and location (quiet, busy). A close selection reduces the confusion that may occur when the labelling is completely uncontrolled [23]. NXP recordings cover five of the most common, everyday acoustic contexts: *bus, car, office, subway and street*.

In addition to audio recordings, the application captures data from all the other phone sensors including pressure, temperature and motion. These information are stored with the aim of combining audio cues with other sensor information. A scheme of NXP dataset recording and deployment procedure is reported here. First, the sound of specific acoustic scene is recorded through an mobile application and sent to a centralized server, together with meta-tags and other sensors information; second, all recordings are listened, analysed

and pruned; third, a model is generated from this data using state-of-the-art techniques and compared with performance on other datasets (e. g. the DCASE 2013 database). To ensure global data collection quality, volunteers have followed precise guidelines:

- **space** – e.g. office recordings are collected in different offices under different conditions (quiet, voices, printer noise, etc.), and in different locations around the world;
- **time** – recordings registered on the same location at different time of the day;
- **system** – using different phones makes the solution robust against channel diversity and microphone quality;
- **user** – recordings from multiple users, because one person recording habits (time, location, etc.) can limit diversity;
- **semantics** – people may interpret differently a context and for this reason a close list of terms (acoustic scene label, meta-tags) are provided within the recording application.

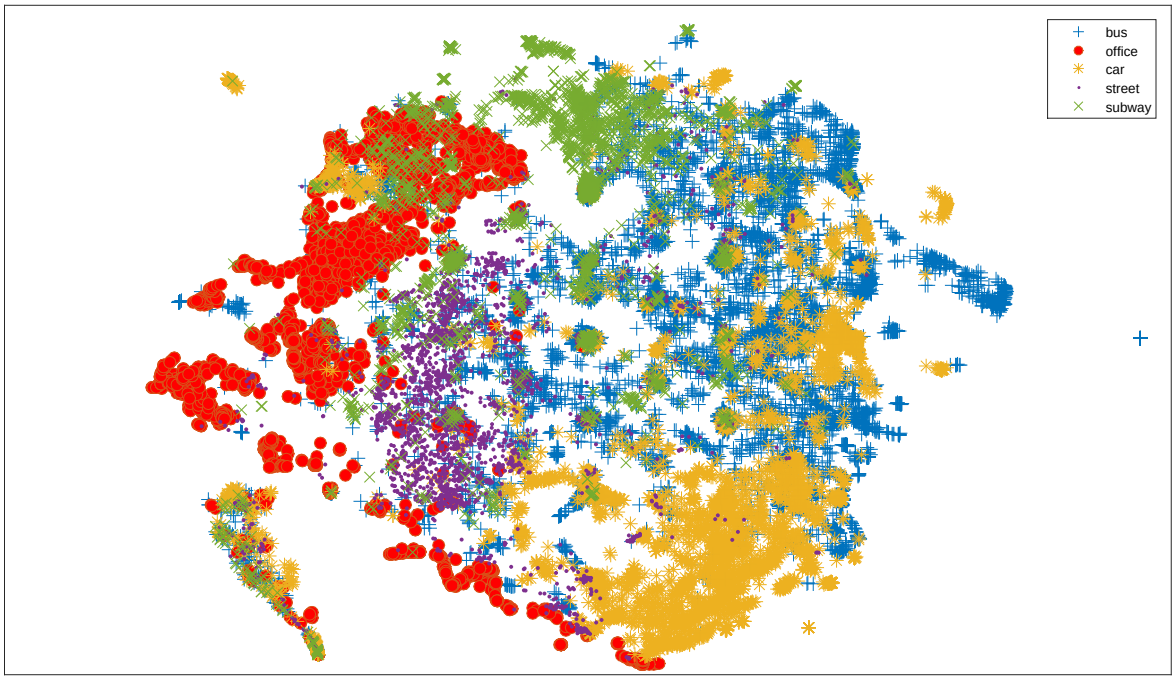
4.3.1 *Feature visualizer for NXP database*

As already done for the DCASE 2013 datasets, visualization techniques are applied to the NXP database in order to represent the underlying data structure. t-SNE visualization combined with Bhattacharyya distance provide an overview of the relationship between scenes and samples. These methods are herein tested on NXP dataset. Features displayed with t-SNE are standard MFCCs, computed according to `MFCC-2000-4s-w.o.C0` system configurations. t-SNE displays data distribution with a perplexity of 50 and a trade-off $\theta = 0.9$. Due to the huge amount of samples, a θ parameter close to 1 prefers a faster estimation to minimal error. Fig. 21 depicts a t-SNE visualisation followed by Bhattacharyya distance-based ranking.

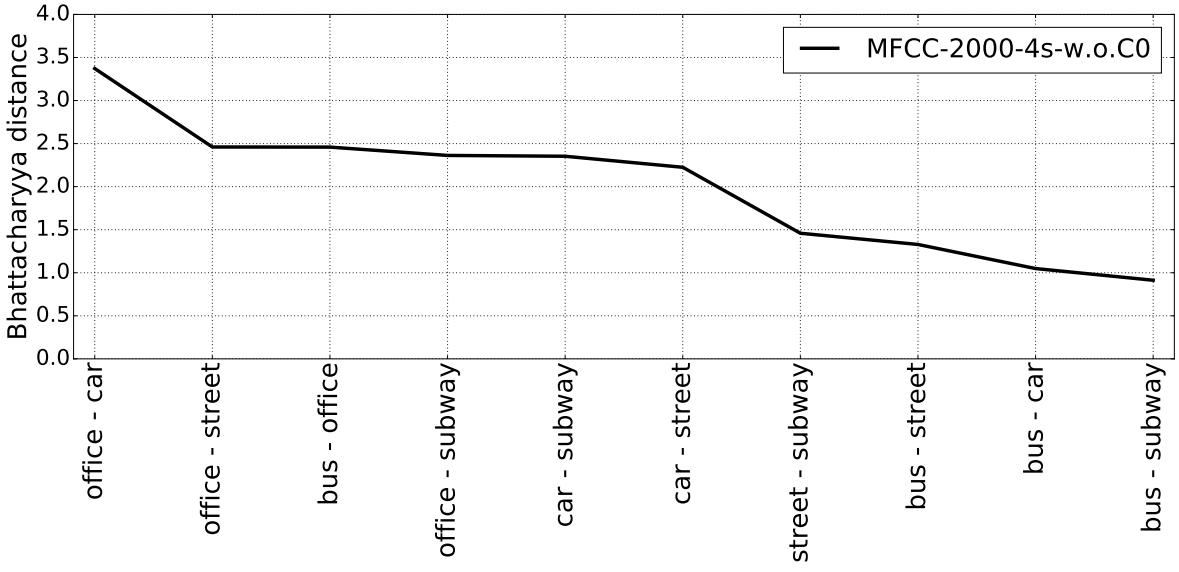
Data visualization shows that some classes are more clustered than others, in particular *office* and *car*. Moreover, the inter-class relationships are highlighted by both t-SNE and Bhattacharyya distance. The *office* environment is the easiest to distinguish whereas *bus-car* or *bus-subway* are the more difficult. The relationship is confirmed from high t-SNE scene overlap and low Bhattacharyya distance: the *transport-like* scenes (e. g. car, bus, subway) are the most inclined to be confused. In contrast to DCASE 2013 results, the t-SNE produces a visualisation which seems more compact and with less overlap even without the coefficient C0.

4.4 APPLYING FEATURE ANALYSIS TO FEATURE DESIGN

Visualisation and feature analysis suggests that current performance are linked to the current class composition of the DCASE 2013 dataset. The same system, trained and evaluated on a different dataset (e. g. NXP dataset), may produce results which are different from the ones already known. The main conclusion of sec. 4.2 is that representing energy is beneficial, but may lead to poor generalisation if condition changes. As an example, different distances between the microphone and the sound source can drastically change energy levels. For this reason, a "global" energy indicator as C0 is here replaced by two "relative" measures: one based on the root mean square (RMS) (RMS-based) and the other on the band energy ratio (BER) .



(a)



(b)

Figure 21: (a) t-SNE visualization of NXP dataset at perplexity = 50 and $\theta = 0.9$. The system is `MFCC-2000-4s-w.o.C0`. (b) Class pairs ordered by their Bhattacharyya distance. A higher value corresponds to a better class separability.

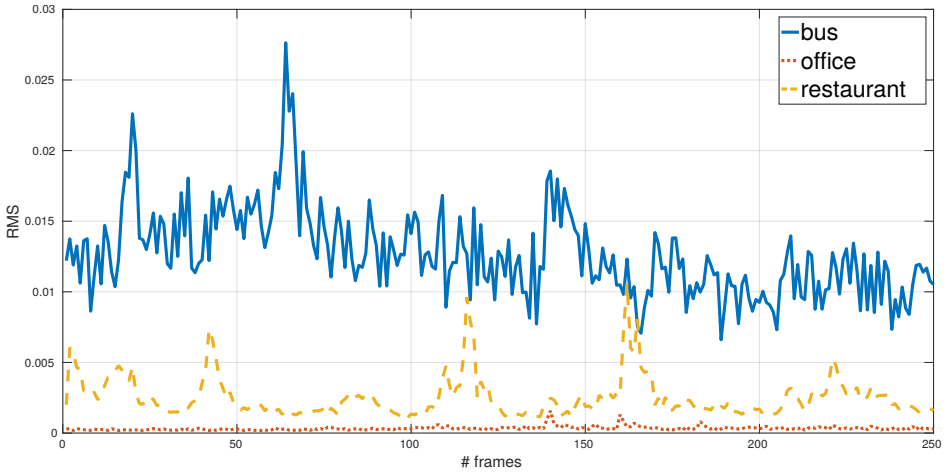


Figure 22: RMS values for 4s segments of 3 scenes from DCASE 2013 evaluation set. *Bus* is depicted in solid line blue; *office* in dotted red line; *restaurant* in dashed yellow line.

4.4.1 *RMS-based features*

The global energy of a signal $x[n]$ can be computed taking the root average of the square of the amplitude, also called the RMS:

$$x_{rms} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x[i]^2}, \tag{21}$$

where n is the number of audio samples in the segment. The entire signal will be represented by successive x_{rms} segments.Fig. 22 presents examples of x_{rms} for a 4s segment for *bus*, *office* and *restaurant* scenes. In this experiment, n is set to number of audio samples contained in 16ms of signal.

For highly stationary contexts (e. g. *office*), the number of times that the RMS value exceeds the average RMS value tends to be small; the presence of impulsive sounds (e. g. loud voices in a *restaurant*) creates distinguishable peaks in the RMS distribution; very noisy contexts (e. g. engine noise of a *bus*) vary even more and their RMS distribution is characterized by a high rate of significant energy peaks.

Influenced by these observations, RMS-based features can be extracted from the RMS distribution. The first dimension of RMS-based feature measures the spread of standard deviation relative to the mean; the second dimension reflects a dynamic range between the highest and lowest RMS peaks. The two feature dimensions x_1 and x_2 are computed as follows:

$$\begin{aligned} x_1 &= \frac{\sigma_{rms}}{\sqrt{\mu_{rms}}} \\ x_2 &= \frac{max_{rms} - min_{rms}}{max_{rms}} \end{aligned} \tag{22}$$

where σ_{rms} and μ_{rms} are the RMS standard deviation and mean relatively to the segment where frame-level RMS are extracted; max_{rms} and min_{rms} indicate the highest and lowest RMS values respectively.

The combination of these two metrics on the DCASE 2013 evaluation set is presented in Fig. 23. Looking more specifically to these features, we notice that *office*, *park* and *quietstreet* have a low variation coefficient (x_1). At the same time, the dynamic range (x_2) of these

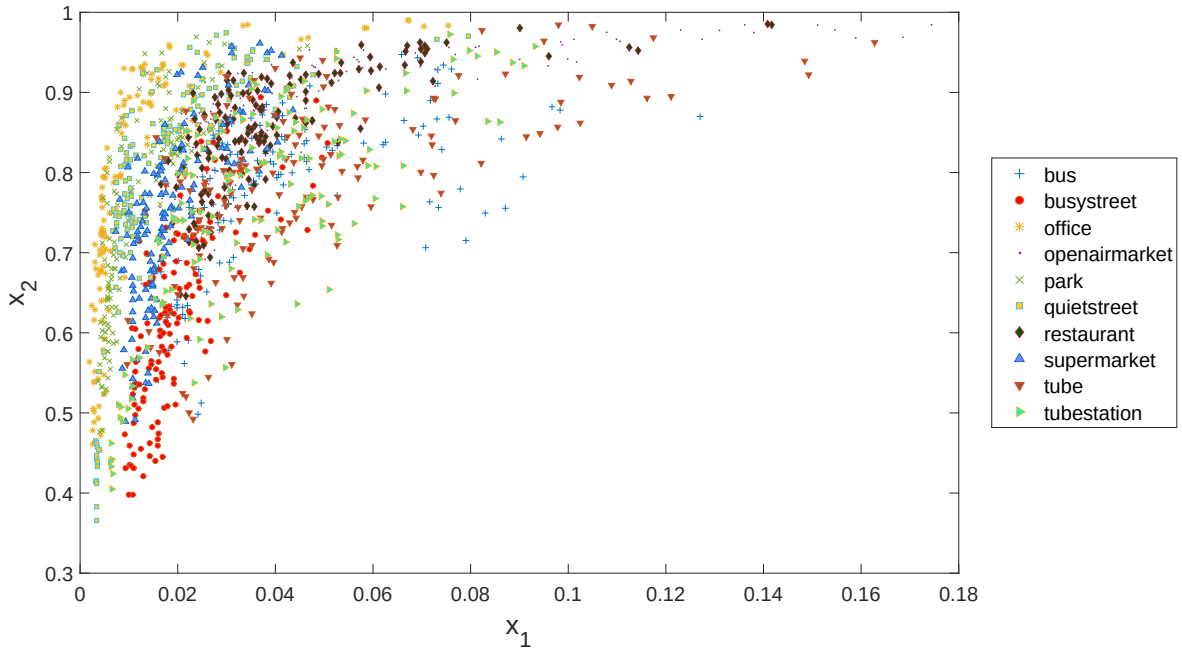


Figure 23: Scatter plot of the RMS-based features. Each point corresponds to a 4s segment of DCASE 2013 evaluation set, represented by x_1 and x_2.

classes varies along the entire $[0, 1]$ range. This is probably due to the presence of speech or impulsive sounds which increase the difference of the max and the min RMS. Other classes such as *bus*, *tube* and *tubestation* are characterized by a high dynamic range and a high variation coefficient. On the contrary, *restaurant*, *supermarket* and *openairmarket* are characterised by their high dynamic range and a low variation coefficient. The use of RMS-based feature expresses the variations of RMS (variance and highest-lowest peaks) as a percentage relatively to the segment where the features are extracted.

4.4.2 *Band energy ratio*

The energy information of a scene can also be expressed as a ratio of the energy present in sub-bands to the total energy. This set of features is referred to as BER. BER features have been adopted in many ASC systems, as described in the literature review of Sec. 2.4. BER features quantify which bands contain the larger portion of energy with respect to the whole spectrum. Features based on BER can be combined with standard MFCCs. The choice of MFCC frequency range is usually critic and BER features are complementary to MFCC features in providing sub-band energy information.

To illustrate the relationship between BER features and acoustic scenes, BER features were computed for the DCASE 2013 evaluation set. The mean BER of each class is depicted in Fig. 24 as a function of 6 sub-bands. At a first sight, sub-bands over 2kHz seem to have marginal discriminatory power. Almost 50% of *bus* energy lies in the $[0, 200]$Hz range; *park*, *office* and *quietstreet* have the flattest BER shape, meaning that spectral energy is distributed over the entire spectrum. *Restaurant* and *openairmarket* show greater BER in the $[500, 1]$kHz band due to high speech components. Similarly, *tubestation*, *tube* and *supermarket* exhibit significant BER between $200 - 1$kHz due to the presence of low-frequency, stationary noise (e. g. locomotives) mixed with voices. *Busystreet*, meanwhile, is mainly concentrated between $[1, 2]$kHz. In fact, *busystreet* samples are a mixture of car engine noise, voices and wind noise.

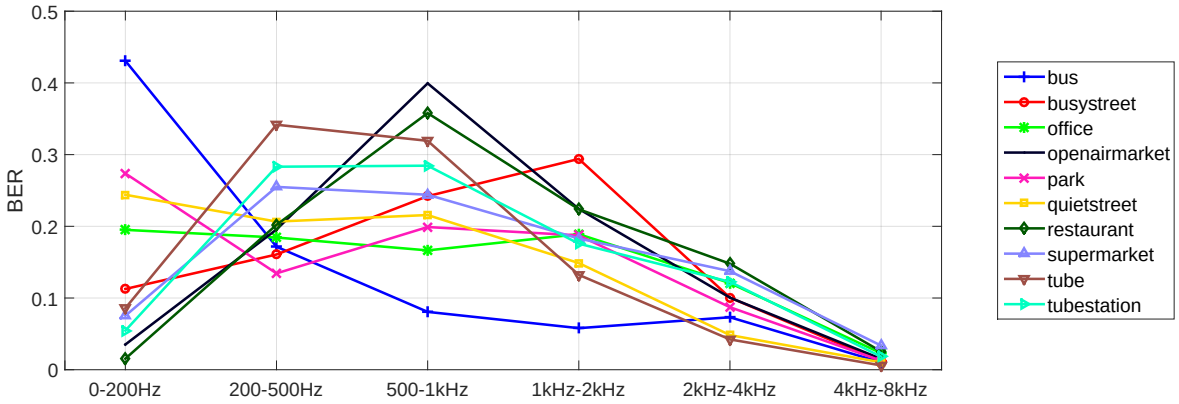


Figure 24: Mean of BER for each class as function of different sub-band ranges. Results refer to DCASE 2013 evaluation set.

4.4.3 *Results & statistical tests*

In line with prior ASC work [6], a Wilcoxon signed rank test has been adopted to determine the validity of the hypothesis that the performance of two distinct classifiers is the same [73]. This test has the advantage of not relying on a particular data distribution assumption. In some cases, the hypothesis of equal performance is valid: one explanation may be that the proposed features do not add significant difference in terms of performance; an alternative explanation may be caused by a modest amount of data where only *macro* differences are observable. In order to better discuss these differences, results are reported for the DCASE 2013 database (development and evaluation sets) and for the larger, non-standard NXP dataset using a 5-fold partitioning into independent training and testing sets. The collection of the internal NXP dataset is a contribution of this thesis and is described in sec. 4.3.

All systems are reported in Tab. 6: `RMS-based` refers to RMS-based features; the BER reflects the ratio of 6 sub-band energy to the total energy of a signal; `MFCC-noC0` denotes MFCCs computed on a larger range $[0, 8]$kHz, without C0. The reasoning behind changing the frequency range from $[0, 2]$kHz to $[0, 8]$kHz stems from the fact that BER features indicates which sub-band contains the most significant contribution so that MFCCs can be extracted over a bigger range of frequencies. The winning system of DCASE 2013 challenge is also reported as `MFCC+RQA-900` system. In contrast to other systems, the latter is applied to longer segment of 30s. All the other systems integrate features over a 4s segment overlapped by 2s. In order to be fully compatible with results in the literature, a majority vote is set up to provide a single prediction for each audio file. A SVM classifier in the form presented in 3.2 is learned in the same way in order to compare all systems.

Results in Tab. 6 show that `RMS-based` features are complementary to MFCCs, passing from 58% to 65% and from 57% to 62%. The `RMS-based` system alone provides a relatively high accuracy for a 2-dimensional feature. Systems including `RMS` and `BER` features achieve good performance on the DCASE 2013 datasets, even though they not reach that of the `MFCC+RQA-900` system. Interestingly, the proposed `MFCC-noC0+RMS-based+BER` achieves the best performance for the NXP dataset, while the winning DCASE 2013 method achieves an accuracy of 79%. This shows that the proposed `RMS` and `BER` features are robust across different datasets, while `MFCC+RQA-900` system behaves well only for the DCASE 2013 dataset, but with poor generalization. A Wilcoxon test with a *p-value* of 5% shows that the best system for each dataset is statistically different from the others.

		Datasets		
Features	**dim**	*DCASE 2013 dev*	*DCASE 2013 eval*	*NXP*
`RMS-based`	2	37% (± 7)	42% (± 13)	28% (± 1)
`MFCC-noC0`	25	58% (± 5)	57% (± 13)	84% (± 2)
`MFCC-noC0+RMS-based`	27	65% (± 6)*	62% (± 10)*	86% (± 2)
`MFCC-noC0+RMS-based+BER`	33	66% (± 10)*	65% (± 8)*	**89**% (± 1)
`MFCC+RQA-900Hz (RNH)`	37	**70**% (± 10)	**76**% (± 5)	79% (± 2)

Table 6: Accuracies computed for different features and different datasets are herein presented. Terms with a * represent results which are not statistically significant according to a Wilcoxon signed rank test with a *p-value* of 5%. The statistical test regards results from the same dataset. `RMS` expresses the amplitude and temporal variance, expressed with a 2-dimensional feature vector; `MFCC-noC0` denotes the MFCCs computed over $[0, 8]$kHz range, without C0; `BER` is the band energy ratios. Composed systems are indicated with the + while with `MFCC+RQA-900Hz`(RNH) is indicated the winning system of DCASE 2013. Best methods of each dataset are indicated in bold.

4.5 FINAL THOUGHTS

This chapter presents the use of a non-linear visualisation technique to map high-dimensional data into a 2 or 3 dimensional space. At the same time, t-SNE can be very helpful in representing the progressive transformations of the input data through deep neural network (DNN) layers; to this end, Chapter 6 shows will see how t-SNE visualises the intermediate data transformations of a neural network. In addition, feature metrics can complement t-SNE visualisations: the *Fisher* score produces a measure of general separability; the Bhattacharyya distance quantifies the level of overlap between class distributions, ranking them from the most easy to the most difficult to separate.

Thanks to this analysis, two main contributions are presented:

- the NXP dataset, a 30h non-standard dataset which serves to cross-validate the performance of the DCASE 2013 database;
- RMS-based features, which capture the variation in RMS values over the segment, and the BER features, which represent the energy in a sub-band as a ratio to the global energy. Results show consistent trends over the public DCASE 2013 datasets and the non-standard but larger NXP dataset.

Data visualisation and feature analysis can identify hidden underlying structure of data. This can help in designing and choosing a set of new features without the need for classification experiments and independently from a specific classifier. To conclude, optimal features are scene-dependent, meaning that depending on the type and nature of acoustic scenes, a set of features may be more or less discriminative.

5

TIME-FREQUENCY PATTERN ANALYSIS

Almost all approaches to ASC utilise traditional features designed predominantly for speech processing applications such as speech or speaker recognition. Even so, experiments in previous chapters showed that these features may not be sufficiently discriminative for the ASC task. Herein are listed the main drawbacks of current ASC systems:

1. **they do not capture both global and local information**. Features determine whether a system represents a generic scene information (such as global energy, spectral envelope, etc.) or whether describes a local-relative variations (as BER, RMS-based features). The use of both global and local information has proven to be effective for ASC literature [34, 45], even though there does not exist a comprehensive approach;

2. **they are based on features not suited to ASC**. For example, MFCCs remain the standard choice in many ASC systems. MFCCs capture only short term variations with minimum dynamic information, whereas correlation in temporal domain may help to discriminate between different scenes. As an example, a promising approach[33] represents complex acoustic structure with features across both time and frequency space. Intuitively, spectro-temporal features should be considered as an alternative to standard MFCC-based approaches;

3. **they imply a temporal structure** even in presence of a sparse and unordered sequence of sounds. In contrast to speech signals, where a strong temporal structure is determined by the phone sequence, ASC is characterized by a comparatively weak temporal structure. Events composing a scene may occur at any time and in any order and duration. As argued in [37], human listeners classify a scene by the presence of a particular sound. This suggests that focusing on the *presence* of certain sounds may improve performance, as reported in [12].

Hence, new features are needed in order to capture complex acoustic structure. This chapter reports work to characterize the distribution of acoustic structure through textural features. The proposed approach is based upon an image processing technique known as local binary patterns (LBP) analysis, which is applied to audio spectrograms in order to capture 'acoustic patterns'. Those capture spectro-temporal structure at sub-band level. To capture specific patterns in each scene, the new features are optionally used to learn a low-footprint codebook of the most frequent patterns. The codebook provides a sparse representation of the acoustic structure. The research hypotheses are that: (i) frequent acoustic patterns can be captured using LBP analysis [87] applied to audio spectrograms; (ii) the new LBP-based features provide complementary information to traditional MFCC features, and that (iii) LBP analysis can be applied as a 'bag-of-features' approach by creating a codebook of the most prominent features and by representing each sample as combinations of these features. Experimental results of LBP-based systems are demonstrated be competitive with the current state of the art. The structure of this chapter is organized as

follows. Sec. 5.1 describes prior works which share with LBP analysis the idea of spectro-temporal patterns. Sec. 5.2 presents LBP analysis and the application to the ASC problem. Sec. 5.3 includes implementation details, experimental results and conclusions.

5.1 PRIOR WORK ON TIME-FREQUENCY PATTERNS

Previous approaches to ASC have focused on selecting and combining standard acoustic features. The literature, e. g. [53, 57, 58, 61], shows that MFCCs are usually the baseline with which other features are combined.

In fact, a study of the human auditory system reported in [37], demonstrated that humans recognise acoustic scenes by mixing audio events with background noise. The correct recognition rate for 19 subjects was found to be in the order of 70% for 25 different scenes. Some events are more probable in some environments and this information enables a scene to be recognised more reliably. Drawn upon these findings, research in [12] concentrates on context-dependent sound events, where histograms of event-occurrences are used to identify the scene (i. e. engine noise occurs more frequently in a *car* or a *bus* rather than in a *office*).

Other approaches based on the modelling of acoustic patterns have also shown their validity. One example is audio motif discovery [88], which uses bio-informatic techniques to find recurrent patterns. Sounds are transformed into a sequence of discrete states, each of them representing a specific audio pattern. A related approach to music genre classification using textural features is reported in [89]. All these methods are characterised by a bottom-up approach, which represents globally an acoustic scene with the occurrence of local acoustic patterns. Temporal recurrence, motif discovery, sound event detection share the notion of acoustic patterns and lend support to the benefit of capturing time-frequency information in a compact way.

5.2 LOCAL BINARY PATTERNS

LBP analysis is a well known approach to feature extraction for automatic face recognition [90]. LBP is an efficient texture operator which labels the pixels of an image (here an audio spectrogram) by comparing their value to those of neighbouring pixels and by representing the result as a binary number. The analysis of acoustic signals using LBP analysis has been reported previously [91, 92] and is applied by treating the spectrogram as a visual representation of the acoustic signal.

The use of LBP for acoustic analysis and feature extraction is motivated by its suitability to texture and structure representation. LBPs are usually used to create histograms which capture the presence of specific patterns. For ASC they provide more discriminative features which reflect the acoustic texture of a scene. The following describes the extraction of raw LBP features, henceforth referred to as `LBP`, and an extension to a *bag-of-features* approach referred to as `LBP-Codebook`.

5.2.1 *System overview*

The new approaches are composed of four stages, as illustrated in Fig. 25:

1. LBP analysis is applied to the spectrogram representation of the full acoustic signal by comparing the magnitude of each time-frequency "bin" to that of its immediate neighbours. The set of raw LBPs are used to generate an LBP histogram which reflects the occurrence of each LBP across the full signal;

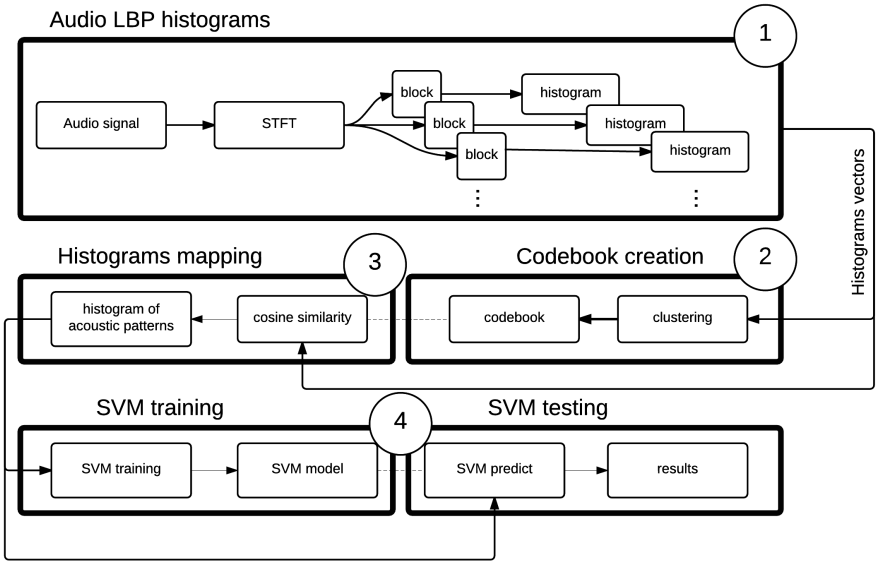


Figure 25: An illustration of the entire system, as explained in Section 5.2.1: **(1.)** LBP histogram generation for each sub-band; **(2.)** Codebook creation, through clustering; **(3.)** Histograms in **(1.)** are mapped to the codebook. This is repeated for each histogram extracted from each block; **(4.)** SVM training and testing by using the histogram of acoustic patterns.

2. histograms are generated for each signal in a large dataset and then clustered to group together the most similar histograms. Resulting cluster centroids are then used to form a codebook;

3. the codebook can be used to map a histogram onto the single, nearest *word* as determined according to a cosine similarity metric. This process results in `LBP-Codeword` features of reduced dimensions which are less redundant and more sparse;

4. ASC is performed using a SVM classifier, applied either to `LBP` ((1.) of Fig. 25) or `LBP-Codebook` ((3.) of Fig. 25) features.

5.2.2 *LBP histogram*

The original idea of LBP is reported in [93]: the operator represents complex textural images in a simple and convenient way through the binary thresholding of the surrounding neighbours of each pixel. Each block around a pixel provides a binary number which expresses the relationships of the surrounding pixels P with respect to the central pixel x_c: if the difference of the neighbours and the central pixel is negative, the result is 0 otherwise it is 1. A histogram h represents the frequency of the binary numbers in each block. The histogram itself expresses the image (or part of it) as the occurrences of binary patterns found in the image.

LBP features have become popular in the image processing community for their invariance to gray-scale changes and their computational simplicity. LBP analysis thus appeals to ASC: the same analysis technique is robust to the presence of spectral-variations; LBPs express global information through the analysis of local blocks; LBPs capture time-frequency information in single feature vectors (i. e. histograms), which better suits a SVM classifier. Standard LBP analysis used in image classification has a block size of 3×3 built around a central pixel x_c with $P = 8$ surrounding pixels. The histogram of $2^8 = 256$ possible patterns is then used as a feature vector to characterise the image. According to the works in [87], by

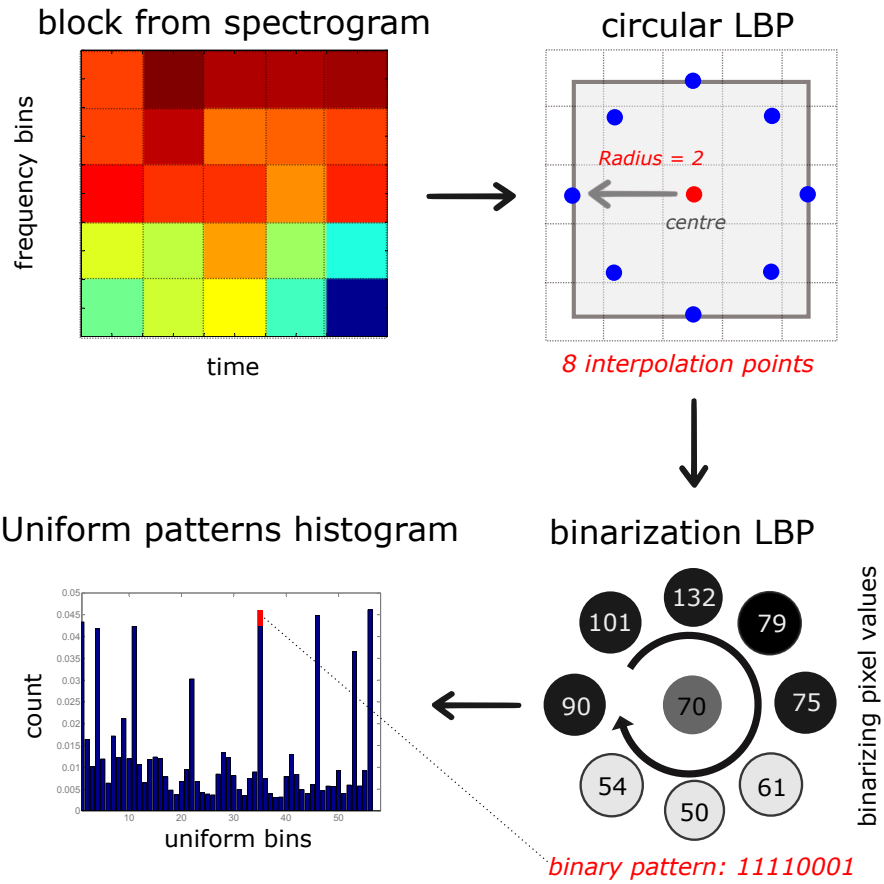


Figure 26: From spectrogram block to LBP histogram: starting from the upper left of the image, the spectrogram block is analysed using $LPB_{8,2}$ with 8 neighbours and radius equal to 2; the local binary code is then generated using Eq. 23; finally the binary code is updated in the corresponding bin of the histogram.

using a circular block with a bilinear interpolation over integer pixel values, it is possible to employ LBP at any radius R with a number of neighbours P:

$$LBP_{P,R} = \sum_{i=0}^{P-1} f(g_i - x_c)2^i, \quad f(x) = \begin{cases} 1, x \geqslant 0 \\ 0, x < 0 \end{cases} \tag{23}$$

where g_i is the pixel value of the i^{th} neighbour obtained with a bilinear interpolation, x_c is the centre of the block, P is the number of neighbours and R is the radius of the neighbourhood. The coordinates of g_i are identified by the pair $(R\cos(2\pi i/P), R\sin(2\pi i/P))$. LBP represent a specific pattern in a compact *code*. This LBP code is computed by multiplying the result of function f (1 or 0) with the power of 2^i and then summing all the values. Thus $LBP_{P,R}$ codes can assume value between 0 and $2^P - 1$. The entire process from a spectrogram block to the histogram is depicted in Fig. 26.

Inspired by the findings in [94], the histogram of patterns is normalized using the L_2-*Hellinger* normalization as $\boldsymbol{x}' = \sqrt{\frac{\boldsymbol{x}}{||\boldsymbol{x}||_1}}$. The resulting normalised feature vector has unit norm $\|\boldsymbol{x}'\|_2 = 1$. It has been shown that this normalization, when applied to histograms, amplifies the discriminative power of LBP features.

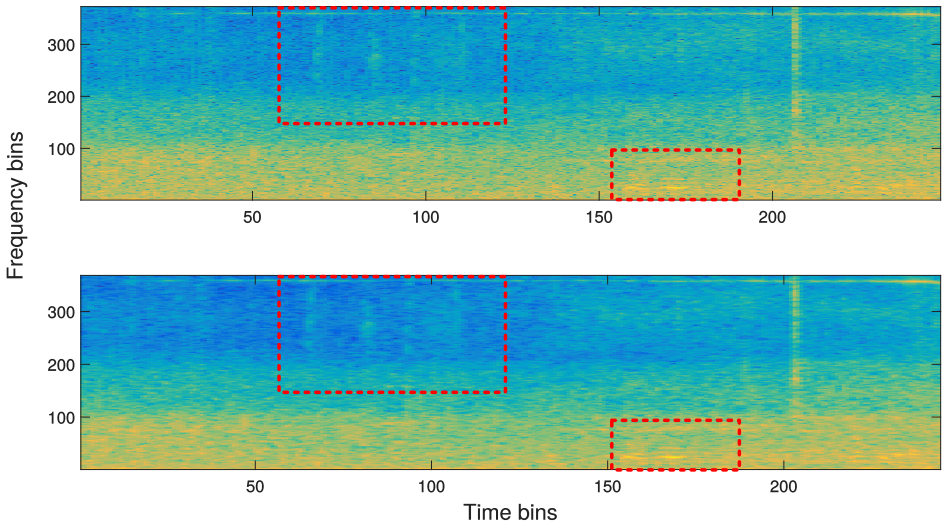


Figure 27: The effect of bilinear interpolation, performed on a *bus* spectrogram in the range $[0, 8]$kHz before the LBP pattern extraction. This interpolation acts as a smoothing operation, which better defines patterns in the spectrogram (e. g. differences in red rectangular areas).

5.2.3 *Application of LBP analysis to spectrograms*

The application of LBP to spectrograms requires some adaptation. Each bin in the spectrogram reflects the amount of energy present in proximity to specific time and frequency bins. Spectrograms, by construction, are characterised by local bin fluctuations (namely bins which can vary significantly in a local area), which may degrade LBP feature representation. LBP is highly affected by fluctuations of bins in the neighbourhood which indeed may change drastically the LBP binary code. In LBP analysis, these fluctuations are rapid transitions in a LBP code from 1 to 0 and vice-versa. Hence, the interpolation of bin values help to attenuate the effect of these fluctuations by globally smoothing the blocks (Fig. 27). Another strategy to add robustness to LBP is to consider only LBP codes for which the number of transitions between 0 and 1 is less than or equal to 2. This subset of LBPs represents the so-called uniform patterns. The remaining non-uniform patterns are often grouped together and considered as a single, distinct non-uniform pattern.

Various modifications to the spectrogram are generally necessary prior to LBP extraction. Experimental works show that analysis of the log-power spectrogram gives better results than the linear-power spectrogram. In addition, bin values are scaled to the range $[0, 255]$ by normalising each single spectrogram:

- I is the power spectrogram of the real part after the application of FFT;
- each bin of the spectrogram I at time-frequency coordinates (m, n) is considered as a *pixel*;
- each pixel $I(m, n)$ is scaled according to

$$I'(m, n) = \frac{\log(I(m, n))}{\max(\log(I))} \times 255 \tag{24}$$

  so that all pixel values will be within the $0 - 255$ range.

The extraction of LBP patterns from I' provides a single histogram. However, identical patterns may occur in different *zones* of the spectrogram. The location of such acoustic patterns, in ASC, produces a meaningful information. Consider two scenes which are

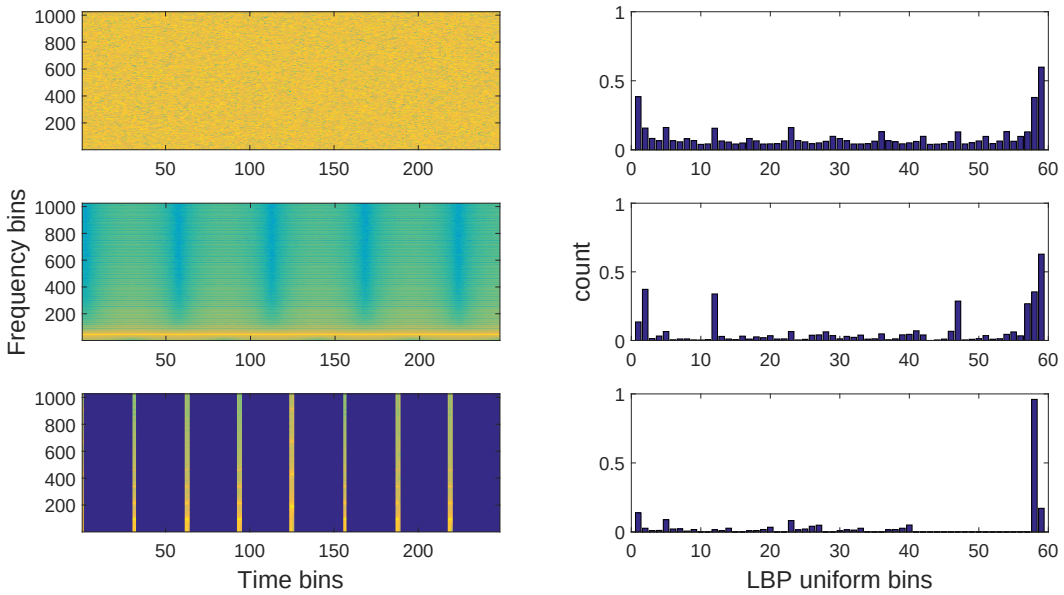


Figure 28: LBP histograms extracted from a white noise (on the right top), a tone at 1000Hz (right middle) and a series of clicks (on the right bottom).

characterised by the presence of tones at different frequencies. If the algorithm does not take into account where LBP patterns come from, the two scenes will be considered similar. In order to model this information, the final LBP features are extracted from spectrograms at sub-band level and then concatenated into a single vector.

5.2.4 *A toy problem*

This section describes the process of feature extraction with a *toy problem* to illustrate how LBPs can be used to differentiate between acoustic scenes. The toy problem is composed of signals of different nature: the first is white noise; the second is a sinusoidal tone at 1000Hz; the third is a succession of 8 equidistant clicks which vertically span the spectrogram. The frequency resolution stems from the application of a FFT with 2048 points. The spectro-temporal resolution of the power spectrogram results in a image of 1025×249 dimension, with a sampling frequency of 44100Hz and a time frame of 32ms overlapped by 16ms over 4s segment. The objective of this toy problem is to illustrate the capture of the most frequent LBPs with respect to the type of acoustic texture they represent. For instance, in the case of impulsive sounds, we expect that the LBPs reflect vertical edges; on the contrary, the constant tone will be captured by horizontal edges.

Fig. 28 depicts the three spectrogram images (left) alongside corresponding $\text{LBP}_{8,2}$ histograms (right). Only uniform patterns with two or fewer transitions between 1s and 0s are captured here. The histogram bin count has been normalized according to a L_2-*Hellinger* normalization. In correspondence of spectro-temporal patterns captured from white noise, the histogram is specially flat, without any evident peaks in the histogram. In contrast, the other two signals exhibit attributes on amount of specific spectro-temporal structure.

The 10 most frequent LBPs are illustrated in Fig. 29 in same order (white noise, tone and clicks) from top to bottom. These patterns represent the spectro-temporal structure captured directly. While patterns of white noise are almost homogeneous (except for maximum one bin), the stationary tone is captured through horizontal edges (visible in patterns $5, 6, 9, 10$). As expected, the impulsive clicks are characterised by edges. These patterns are captured with LBPs $6, 8, 9$. In general, homogeneous patterns (i. e. all 1s or 0s) are the most frequent in the three examples.

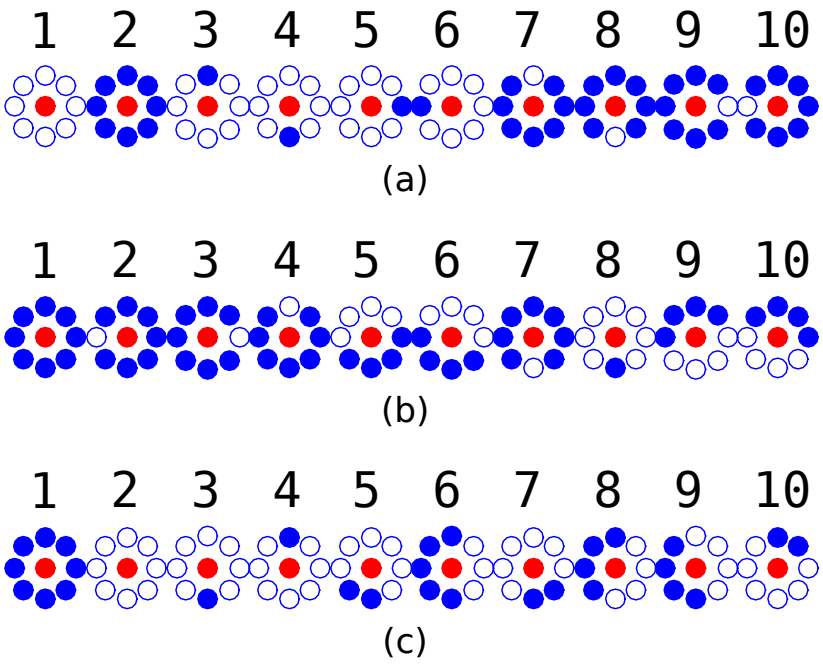


Figure 29: Starting from the left, each column represents the most occurring LBP patterns for (a) white noise, (b) sinusoidal tone and (c) clicks signals. The circular LBP has 8 equidistant points, with a radius of 2. The red dot indicates the centre of each pattern whereas the empty/full blue dot corresponds to 0/1 in the binary pattern.

5.2.5 *Codebook creation*

In order to make the representation more compact, a bag of features (BoF) approach is applied to LBP features. The principal idea is to extract automatically, via unsupervised k-means clustering, a *codebook* of the most representative histograms. Each acoustic scene is then represented as a distribution of the *words* gathered in the *codebook*. This method is based on the well-known BoF technique, popular in image retrieval tasks [95]. The spectrogram of each test sample is represented in terms of the distribution of codebook words whose distance to the closest word is determined according to a cosine similarity metric. The cosine distance is well suited as a metric for histogram features [96]. The BoF method optimizes the aspects of LBP and codebook creation which are pertinent to ASC problem: the local descriptors represent spectrogram clips as a stack of local properties computed over smaller blocks; the codebook is a way of representing the entire spectrogram as a set of local descriptors. In fact, LBPs capture local time-frequency properties of a scene and the codebook represents each recording as a combination of codebook words. As illustrated in Fig. 30, these words play the role of sounds events and each scene can be represented as a collection of these *sound words*.

The main steps of codebook creation are as follows:

1. LBP histograms are extracted from short time-frequency representations, split into sub-bands. This helps to identify patterns not only from their shape but also for their frequency;

2. a codebook of k-words is extracted using k-means clusters for LBPs from each class;

3. LBPs histograms are then mapped to the closest word in the codebook using the cosine distance. A higher-level histogram represents the full recording as a combination of codebook words;

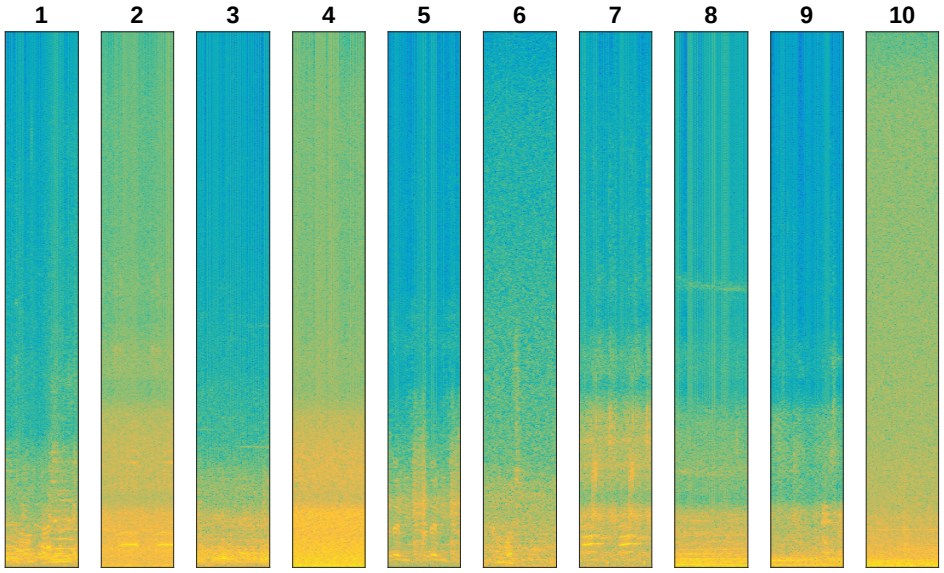
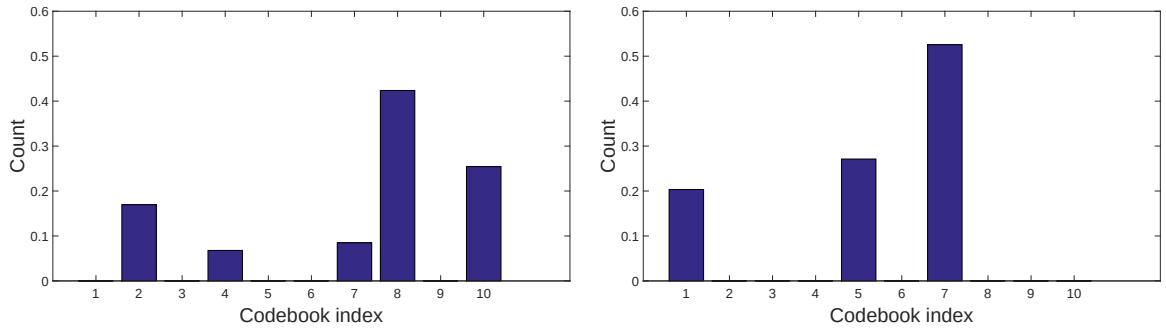


Figure 30: The corresponding log-power spectrogram excerpts of 1s selected with respect to the closest LBPs cluster centroids. The sampling frequency of the selected signal is 44100 and the dataset is the DCASE 2013 evaluation set.



(a) (b)

Figure 31: The codebook histograms for a *bus* scene (a) and a *restaurant* scene (b) for the DCASE 2013 evaluation set. The codebook words are depicted in Fig. 30

4. the `LBP-codebook` features are used alone or in combination with standard features (e. g. MFCCs) before classification.

An example codebook of 10 words is displayed in Fig. 30. These spectrogram clips correspond to the audio spectrogram closest to the cluster centroids. Words 1, 5 and 7 comprise mainly voices (indistinct chatting, single male voice and child voice); a beep sound is captured in word 2; white noise-like clips are represented by words 4 and 10; finally words 3, 8 and 9 correspond to engine (acceleration or stationary) noise. Examples of codebook composition in real acoustic scenes are depicted in Fig. 31, where the codebook words reflects those of Fig. 30. Interestingly the *bus* sample, to the left inset of Fig. 31, is identified mainly by word 8 (engine noise); *restaurant* sample is represented by the presence of words 1, 5 and 7 (speech/voices clips).

The size of codebook depends on the parameter k, which expresses the number of clusters better representing the dataset: when k is too small, the codebook is not sufficiently discriminative to distinguish between different scenes; when k is too large, the resulting histogram is less sparse and therefore less generic. The optimal k for the DCASE 2013 development dataset was found to be 50, as reported in Tab. 7. The same k is supposed to also represent the evaluation set.

Table 7: The accuracy and confidence intervals (± CI) for DCASE 2013 development dataset as a function of codebook sizes obtained with a k-means clustering. In bold the best results.

Codebook size k	10	30	50	70	100
DCASE 2013 dev	47% (± 6)	52% (± 6)	**55**% (± 5)	50%(± 5)	54% (± 7)

5.3 EXPERIMENTAL RESULTS OF LBP SYSTEMS

This section describes datasets, protocols, implementation details and metrics. In the last part, results are provided followed by a discussion of the advantages and limitations of LBP approach.

5.3.1 *Datasets & protocols*

The LBP algorithm is evaluated using 4 databases with a diverse composition of scenes and recording conditions. This evaluation is therefore more focused on cross-database performance. A fair evaluation involves several databases to test the capacity of a method to generalise to new conditions: a relative performance metric across different datasets will assume more significance than an absolute level of performance on a single dataset. The four databases used for experiments reported in this chapter are: the DCASE 2013 development set, the DCASE 2013 evaluation set, the NXP dataset (see Sec. 4.3) and the Rouen dataset. The Rouen dataset is a recently released public dataset [33]. It comprises about 25 hours of recordings of 19 scenes, registered with smartphones to reflect a potential ASC scenario. This dataset has a sampling rate of 22050Hz with 30s recordings similar to the DCASE 2013 database. For the first three datasets, a 5-fold partition was used to separate training and testing data. The Rouen dataset, instead, has a different standard protocol based on a 20-fold partition [33].

5.3.2 *Implementation details*

Baseline features are the same as described in Sec. 4.4. These are used with the `MFCC+RQA-900` system analysed in Chapter 3. `LBP` features are extracted from 4s audio segment of acoustic signals, using 8 neighbours with a radius equal to 2. LBPs are extracted from log-power spectrogram segments which is first split into 3 sub-bands (900Hz, 2kHz, 8kHz), with the aim of distinguishing between similar patterns coming from different spectral sub-bands. The spectrogram has a time resolution of 32ms overlapped by 16ms. Histograms of 59 bins (58 uniform patterns plus 1 bin grouping all the non-uniform ones) are extracted separately for each sub-band and concatenated to form a single feature vector. The resulting histogram is normalised with the L_2 *Hellinger* normalization.

`LBP-Codebook` features stem from LBP analysis applied to smaller 1s segments with an overlap of 0.5s. Clustering is applied to obtain 50 clusters for the DCASE 2013 evaluation dataset and 100 for the larger NXP and Rouen datasets. Experiments for finding the best k cluster value are performed on the development set of DCASE 2013 and on training set of NXP and Rouen databases. These values were found to be optimal given the different dataset sizes. `LBP-Codebook` features extracted from each sub-clip are aggregated over the file duration to obtain a single BoF histogram per recording. `LBP-codebook+MFCC-900Hz` refers to the combination of the proposed system with standard MFCCs.

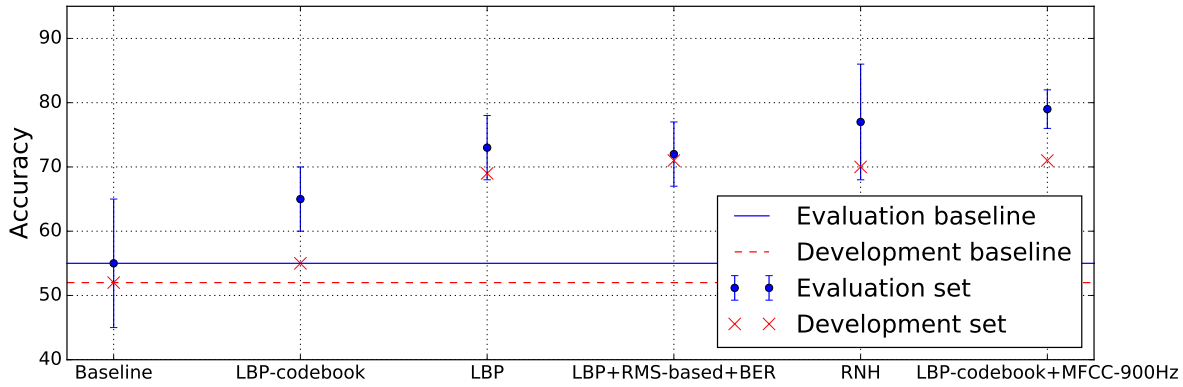


Figure 32: Plot shows the accuracy mean with 95% confidence intervals (CI) over 5-fold cross-validation for DCASE 2013 dataset. In blue circles the values of evaluation set, whose baseline is expressed also with a blue line; in red stars the values of the development set with the baseline expressed in dashed red line. Except for the baseline and the RNH, the other systems have been proposed in this work.

5.3.3 *Results*

The proposed LBP-based systems are compared with state-of-the-art systems in Fig. 32. `LBPs` achieves an accuracy of 73%, performing 18% better than the MFCC baseline. In addition, replacing RQA in the RNH system gives an accuracy of 79%. LBP combined with energy-based features (BER and RMS-based) achieves an accuracy of 72%.

The multi-dataset evaluation is reported in Tab. 8. The statistical significance is assessed through a Wilcoxon signed rank test. At first sight, `LBP`-based features outperform `MFCC-noC0` features for all datasets. Adding `RMS-based` and `BER` features to `LBP` further increases performance, reaching the best accuracy for the DCASE 2013 dev, NXP and Rouen datasets. For the larger datasets such as NXP and Rouen, the configuration `LBP+RMS-BASED+BER` achieves 93% and 88% respectively.

In particular, for the Rouen dataset, results are comparable to the 87% accuracy reported in [33], which employs an image processing techniques applied to a time-frequency representation. The `MFCC-RQA-900` system remains the second best system for the DCASE 2013 dataset (eval) but it generalises poorly to other datasets. Surprisingly, for NXP and Rouen datasets, the addition of RQA to MFCC features has no impact. The `LBP-codebook` system achieves an accuracy of 90% for the NXP dataset whereas when combined with MFCC improves still further performance, reaching the highest accuracy achieved for the DCASE 2013 evaluation set. Finally, the `LBP+RMS-based+BER` system seems to be the most consistent across the four datasets.

LBP analysis extracts patterns from the comparison of central pixel with its neighbours. This property suggests that the proposed LBP system should be less affected by energy variations. In that sense, several volume gains are applied on the testing recordings to prove the energy-invariance property of the proposed systems. Gains in the range $[-12, -6, 0, 6, 12]$ dB are applied to the signal amplitude. Curves plotting accuracy as a function of the gain are depicted in Fig. 33. As expected, `LBP` and `LBP+RMS-based+BER` systems seem the less impacted by energy changes, showing the best trade-off between accuracy and robustness. The figure also shows that the replacing of RQA features with LBP-based codebook features is beneficial not only in terms of absolute accuracy, but also in robustness to different energy.

		Datasets			
Features	dim	DCASE 2013 dev	DCASE 2013 eval	NXP	Rouen
MFCC-noC0	25	58% (± 5)	57% (± 13)	84% (± 2)	79% (± 1)
MFCC-noC0+RMS-based+BER	33	66% (± 10)	65% (± 8)	89% (± 1)	85% (± 1)
MFCC+RQA-900Hz (RNH)	37	70% (± 10)*	76% (± 5)*	79% (± 2)	79% (± 1)
LBP-codebook	50-100	55% (± 5)	65% (± 5)	90% (±3)	70% (± 1)
LBP	177	69% (±6)*	73% (±5)	86% (± 1)	84% (± 0.5)
LBP-codebook+MFCC-900Hz	76	**71**% (± 7)*	**79**% (± 3)*	92% (±2)	85% (±1)
LBP+RMS-based+BER	185	**71**% (± 7)*	72% (± 5)	**93**% (± 2)	**88**% (± 1)

Table 8: Accuracies computed for different features and different datasets are herein presented. Terms with a * represent the best results which are not statistically significant according to a Wilcoxon signed rank test with a *p-value* of 5%. The statistical test regards results from the same dataset. `RMS` expresses the amplitude and dynamic variance, expressed with a 2-dimensional feature vector; `MFCC-noC0` denotes the MFCC computed over $[0, 8]$kHz range, without C0; `BER` is the band energy ratios. `LBP` indicates the textural features over three sub-bands spectrogram. Composed system are indicated with the + while with `MFCC+RQA-900Hz` is indicated the current state-of-the-art. We indicate the best method of each dataset in bold.

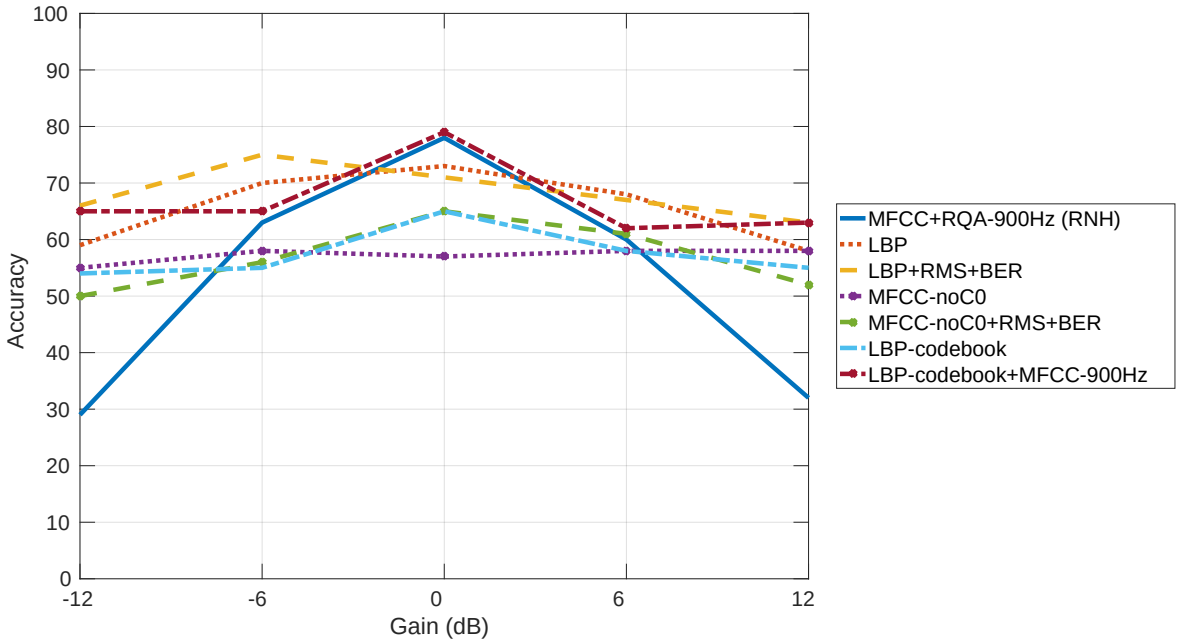


Figure 33: Accuracy as a function of gains (dB) applied to the testing samples (at 0dB no gain has been applied). The reference dataset is the evaluation set of DCASE 2013.

5.3.4 *Main limitations of LBP approach*

This chapter proposes a promising, new approach to feature extraction for ASC. LBP features capture the audio structure and are complementary to conventional MFCCs. Their combination competes with the performance of recurrent quantification analysis (RQA), adding further weight to the benefit of capturing textural features of complex acoustic structure. In addition, a bag-of-features approach is shown to reduce feature dimensionality while still improving on baseline performance. With reduced computational complexity, the codebook approach is perhaps better suited to scenarios with a limited set of resources.

The recent availability of larger datasets (NXP, Rouen) enables new approaches to be evaluated on a broader set of scenes, conditions and type of classes. The capacity of a system to perform well on heterogeneous conditions assumes a key role determining the most reliable technology. This chapter shows that the combination of LBP with RMS-based and BER features provides the most consistent results across different datasets. Moreover, the proposed system is more robust to energy variations while maintaining a constant level of performance.

The codebook could also be trained using a larger pool of readily available data in order to recognize distinct acoustic events rather than abstract time-frequency patterns. This approach may facilitate the learning of codebooks for the same distinct events, e.g. car horns, or engine noise) which may be beneficial, especially if these events are learned in a discriminative framework tailored to the scene classification task.

Albeit providing the best performance reported here for this thesis, LBP-based features have two main limitations:

- the localization of the LBP patterns is partially lost when the histograms is built. Different to images, patterns extracted on the upper part of the spectrogram have a different spectral *meaning* than the same patterns in the lower part. This has been partially solved by splitting the spectrogram into 3 sub-bands and then extracting 3 separate histograms. Nevertheless, localisation of LBP patterns in the spectrogram is specific to audio analysis and should be better represented;

- features are still hand-crafted, necessitating significant effort to choose the best configuration. Among them, the shape of the LBP, the type of uniformity patterns, the operations performed on the spectrogram, the number of sub-bands, the size of the codebook, etc. All these parameters have to be manually optimised.

With the variability in acoustic scenes being so high, the design and tuning of LBP features is still a long and complex process which is exacerbated by the number and type of classes. Every time a new class is introduced, features will likely require re-optimisation. Even so, the LBP approach is a step beyond the use of traditional features for ASC and also a new direction towards the capture of spectro-temporal structure in acoustic scenes.

6

A DEEP LEARNING APPROACH

Acoustic scenes usually exhibit a high degree of variability, both inter-class and intra-class. Because of this variability, ASC is a particularly challenging statistical pattern recognition task. Almost all current approaches rely on hand-crafted features chosen specifically to facilitate discrimination between an often small set of known acoustic classes. The previous contribution in this domain, which investigated the use of local binary pattern (LBP) features, is no exception.

With the variability in acoustic scenes being so high, the premise of the research presented in this chapter is that hand-crafted features are a bottleneck to ASC performance and that automatically derived features have greater potential.

Deep learning techniques have brought tremendous advances in a huge range of different statistical pattern recognition applications [97] and is now the state-of-the-art in many, if not the majority. These techniques and tools offer one alternative to hand-crafted features and a suite of different approaches to automatic feature learning from complex input data (e. g. images and audio).

This chapter reports a first attempt to harness the power of automatic feature learning for ASC using deep learning architectures known as convolutional neural networks (CNNs). Given the promising results obtained with image processing techniques applied to spectro-temporal inputs (e. g. LBP in Chapter 5 or histogram of gradients (HOG) [33]), CNNs seem to be a natural candidate to avoid reliance on hand-crafted features. Experiments with deep learning architectures were made possible by the release of larger databases, namely the DCASE 2016 database. Together with the data, a public evaluation was also announced. Hence, the system proposed in this chapter is compared with other systems. As for the previous edition in 2013, the DCASE 2016 public evaluation significantly advanced the ASC domain.

Experimental results reported in this chapter confirm that the level of performance betters that obtained with hand-crafted features. This can be achieved with deep learning methods and automatically derived features. Furthermore, the property of automatic feature learning is particularly desirable in a task as ASC where the acoustic variability is a predominant factor. The remainder of this chapter is organized as follows: Sec. 6.1 summarizes the prior application of deep learning architectures to the ASC problem; Sec. 6.2 presents how CNN was adapted to the ASC task; Sec. 6.3 describes specific implementation details and experimental results; Sec. 6.5 provides a qualitative evaluation of the features learned by the network; conclusions and discussions are presented in Sec. 6.6.

6.1 PRIOR WORKS ON DEEP LEARNING APPROACHES

The ASC literature shows that the majority of ASC approaches utilise features developed for other related tasks such as speech or music genre recognition (literature review in Chapter 2). Recent works explored the use of features which capture time-frequency correlation. Some of these works draw upon methods popular in other 2-dimensional domains such as image

processing. LBP, for instance, represents an audio spectrograms with a histogram of the most occurring patterns [98]. Similarly, HOG encodes the direction of variations in CQT-based spectrograms [33].

Having been applied so successfully to other related problems, deep learning techniques [6] are now emerging [99]. Deep neural networks (DNNs) are able to identify and extract optimized, discriminant features from training data and thus offer one alternative to hand-crafted features. Many different architectures and data input representations have been investigated for a host of different applications such as image and speech recognition [100, 101].

While the first investigation of DNN approaches to ASC [99] showed promising results, the work was based upon MFCC features. Thus, the potential benefit of deep learning was thus still curbed by the initial use of MFCCs.

This chapter reports the experimental works with a particular approach to deep learning involving convolutional neural networks (CNNs). The main reasons for this choice are (i) the possibility to replace hand-crafted features with features learned automatically and (ii) the possibility to use time-frequency representations as inputs to the network, in line with previous research on spectro-temporal LBP patterns.

6.2 THE PROPOSED CNN ARCHITECTURE

The motivation behind the application of CNNs to spectro-temporal inputs are biologically inspired. As studies on animal auditory cortices confirm [102], temporal and spectral properties of sound contribute to an unified, correlated descriptor. In other words, features which encode time and frequency together better suit to the biological representation of a sound. Therefore an useful aspect of automatic learning is its applicability not only to a frame-wise spectral content, but also to the *concatenation* of consecutive frame-wise spectral contents. Standard CNN architectures require significant adaptation to capture the ASC peculiarities. A deep architecture is a stack of connected layers, each of them performing a specific operation (e. g. input layer, convolutional layer, etc.). Each layer has a specific role in relation to the ASC task and is described in further details in the following.

6.2.1 *Global structure*

CNNs have a multi-layered, deep network architecture. Differently from MFCCs which decorrelate the data with the DCT, CNN takes as input the log mel-filtered spectrogram mimicking an image processing behaviour. In the convolutional layer, each hidden unit is not connected to all the inputs from previous layer, but only to an area of the original input space, called *receptive field*. These small parts of the whole input space are connected to the hidden units through the weights $\boldsymbol{w}$ and bias $\boldsymbol{b}$. This operation is equivalent to a *convolutional filter* processing. The architecture proposed in this work is illustrated in Fig. 34. It is composed of an input layer, a stack of convolutional and pooling layers, a fully connected hidden layer and a final output layer.

CNNs rely on *convolution* and *pooling* operations: the convolutional layer convolves a set of filters over portion of the input whose filters are shared among the entire input space; the pooling can be seen as a *down-sampling* operation which focuses more on the pattern itself rather than the exact location in the input. This adds robustness to small modifications and translations in the input space. A *deep* architecture replicates these operations in a stack. In this way, the filters at each layer capture patterns at higher level of abstraction, because they work on lower resolution inputs coming from the pooling layer. Eventually, the fully connected layer connects units coming from all local positions to perform a global

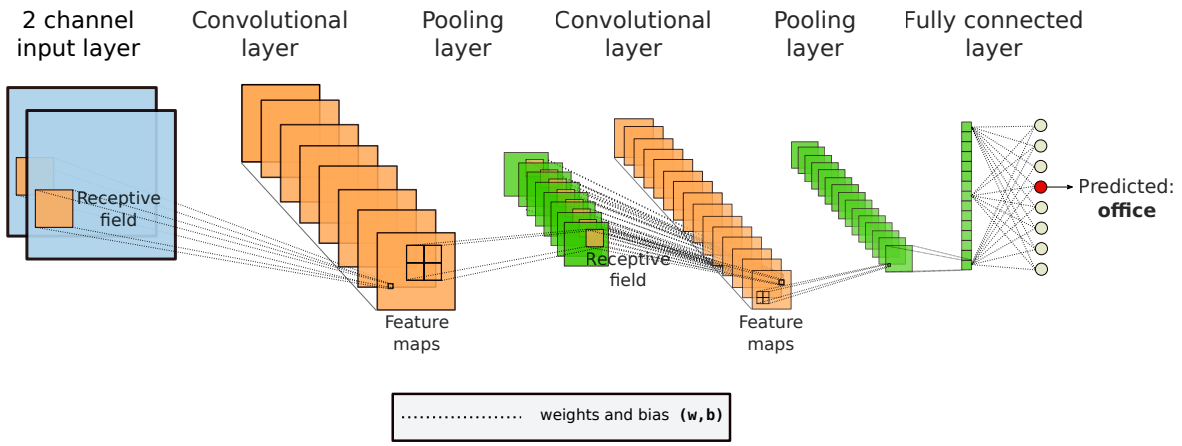


Figure 34: An example of CNN architecture investigated in this work: the input is a 2-channel static and dynamic spectrograms. These are followed by two, stacked convolutional and pooling layers. Fully connected and output layers produce the probabilities of the input data belonging to each acoustic class.

classification of the input. As for LBP histogram in 5.2, the initial spectrogram inputs are represented by the combination of their local components.

6.2.2 *Input layer*

In the proposed CNN architecture, the spectro-temporal inputs are obtained through spectrogram operations. These inputs are image-like spectrograms, with the rows representing the frequency bins and columns the temporal frames. Frequency bin are reduced in dimensions by using a logarithmic mel-scale filters bank. Due to this operation, the level of spectral resolution is mainly preserved whereas the frequency dimension is reduced.

In the application of CNNs to computer vision tasks, input images are typically represented with colour channels (e. g. red, green and blue) [103]. In the proposed CNN, this same idea is adopted in the application of CNNs to ASC. As illustrated in Fig. 35, inputs take the form of (i) a static, log-Mel spectrogram and (ii) a separate, dynamic spectrogram representation composed of its first derivative parameters (Δ). These two representations form a two-channel input ($D = 2$) to the network so that hidden units can combine static and dynamic information. The input of the CNN have three dimensions: the width (L), the height (A) and the depth (D), the latter being equal to number of channels [100].

In addition, as illustrated in Fig. 35, spectrograms are segmented into shorter segments which are treated as independent inputs. For inputs with $D > 1$, the segmentation is applied in the same way.

6.2.3 *Convolutional layer*

Complex acoustic scenes contain discriminative spectro-temporal structure, e. g. *engine noise* or *telephone ring-tones*. These characteristics are referred to as *local patterns*, namely a correlated concentration of energies over both frequency and time. Engine noise, for example, is characterized by a predominant local pattern spanning the time dimension whereas *ring-tones* may exhibit a highly harmonic pattern spanning the frequency dimension. Experimental results with LBP patterns reported in Chapter 5 confirm that the capture of *local* patterns is beneficial to ASC task: a scene is therefore represented by the presence of local spectro-temporal patterns rather than by the global characteristics.

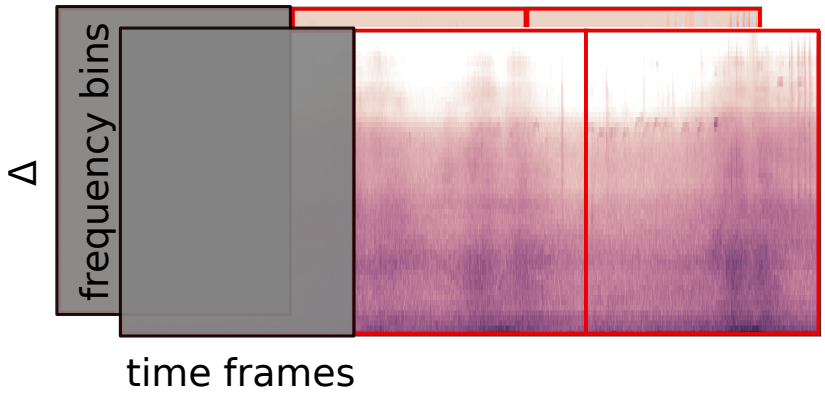


Figure 35: CNN input data is a pair of static (log-Mel) and dynamic (first derivatives, Δ) spectrograms. Both static and dynamic spectrograms are segmented into shorter segments, treated as independent inputs. Each input is characterised by its width, height and depth.

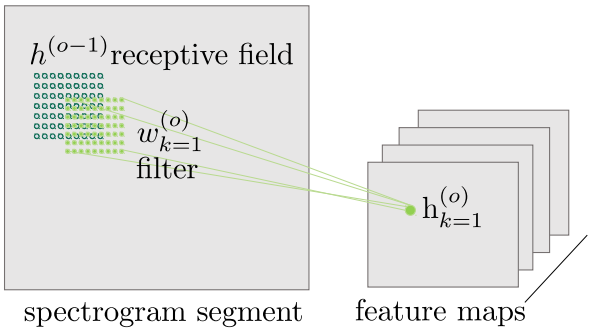


Figure 36: The input ($h^{(o-1)}$) of the convolutional layer is convoluted with the first kernel $w^{(o)}_k = 1$ resulting in the hidden unit of the first feature map $h^{(o)}_{k=1}$.

This *locality* concept is replicated also in the convolutional layer. In the proposed CNN architecture, each neuron in the convolutional layer is connected to a local region (including its depth). This local region is referred to as *receptive field* making the *locality* concept determinant in the construction of the convolutional layer.

Each receptive field of the previous layer is connected to the activation output of the current layer through weights $\boldsymbol{w}$. All convolutional neurons which share the same $\boldsymbol{w}$ and biases $\boldsymbol{b}$ (referred to as *filters*) are grouped together under the name of a *feature map*: each feature map is the output of a set of shared weights applied to the previous layer.

As an example, the initial convolutional layer which has a raw spectrogram segment as its input is depicted in Fig. 36. This figure illustrates the receptive field $h^{(o-1)}$, the first filter and the corresponding first feature map. The depth of the input spectrogram is equal to 1 ($D = 1$) and the total number of feature maps is 4. In the next convolutional layer, D will assume the number of the previous layer feature maps.

Weights and biases are shared among the entire feature map in order to reduce the total number of trainable parameters. A given filter is applied to all receptive fields of the previous layer with a specific *shift* from one position to another. Each position results in an activation of the neuron and the output is collected in the corresponding feature map.

Receptive fields, filters and feature maps characterise the activation function of the $(o)^{th}$ convolutional layer with respect to the $(o-1)^{th}$ previous layer as follows:

$$h^{(o)}_k = f\left(\sum_{d=1}^{D}\sum_{l=1}^{L}\sum_{A=1}^{A} w^{(o)}_{d,l,a,k} h^{(o-1)}_{d,l,a} + b^{(o)}_k\right), \tag{25}$$

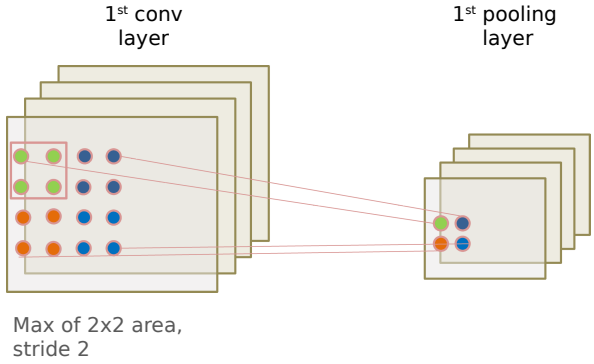


Figure 37: The pooling is applied to 2×2 blocks of the feature maps, with a stride (or shift) of 2 bins. In this example, for each of the 2×2 blocks represented in the convolutional layer, the maximum value is taken and represented in the pooling layer.

where $h_k^{(o)}$ indicates the hidden neuron of the k^{th} feature map of the o^{th} layer. The activation function is f. The resulting hidden neuron is connected to the receptive field $h^{(o-1)}$ of area $L \times A$ through the corresponding convolutional filter w. L, A and D are namely the width, height and depth of the filter.

Hidden unit outputs $h_k^{(o)}$ form new layers of spatially connected neurons which are referred to as *feature maps*. Different *feature maps* can be formed from combinations of locally connected hidden units, each sharing the same weights applied to different positions of the input space.

6.2.4 *Pooling layer*

Pooling layers are applied to the outputs of the convolutional layer in order to reduce their resolution. Different strategies can be applied. Among the simplest is a *max-pooling* operation whereby a block of values in the pooling layer input are replaced with their single, maximum value. What is effectively an operation of *downsampling* is shown to not only reduce dimensionality, but also to provide invariance to translation in the input [104]. An example of *max-pooling* operation is displayed in Fig. 37.

For the ASC task, *max-pooling* produces invariance to small changes in spectro-temporal structure. For example, the same local pattern (e. g. engine noise) centred on a specific frequency may vary only marginally from one recording to another. The pooling operation reduces the dependency over frequency or time resolution, giving more importance to pattern structure rather than spectro-temporal locations.

6.2.5 *Fully connected layer*

Convolutional and pooling layers are replicated in sequence, by using the output of the pooling layer as the input of the convolutional layer. As the architecture becomes at deeper level, the receptive fields will represent bigger areas of the input. This produces higher-level representations of the input data at every layer.

Final outputs are obtained with a fully connected *softmax* layer which produces a score for each class. Inputs to the fully connected layer are outputs emerging from the last convolutional or pooling layer. By definition, fully connected layers do not operate at a local level and instead represent the entire input as combinations of local patterns.

6.2.6 *Learning the network parameters*

The loss function l used for CNN training (i. e. for the optimization of model parameters $\theta = \{\boldsymbol{w}, \boldsymbol{b}\}$) measures the error between a target y_n and predictions $\hat{y}_n$:

$$l(y_n, \hat{y}_n) = y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n). \tag{26}$$

In fact, the predicted output $\hat{y}_n$ is obtained by successive operations involving all parameters $\theta = \{\boldsymbol{w}, \boldsymbol{b}\}$. These parameters are not optimised based on the n^{th} sample. A cost function $J(\theta = \{\boldsymbol{w}, \boldsymbol{b}\})$ refers, instead, to the average loss over N training samples:

$$J(\theta) = \frac{1}{N} \sum_{n=1}^{N} l(y_n, \hat{y}_n). \tag{27}$$

The optimal set of parameters $\theta = \boldsymbol{w}, \boldsymbol{b}$ are those that minimises the cost function J. This can be also seen as adjusting the convolutional filters depending on the predictions $\hat{y}_n$ over the training set. In other words, CNNs create convolutional *filters* which minimise the cost function. The standard algorithm used to solve the minimisation problem $\arg\min_\theta J(\theta)$ is the Gradient descent (GD), which finds the local minimum of a function by iterative optimization steps. At each iteration of the GD algorithm, the optimal directions is obtained by the negative of the gradient of $J(\theta)$.

When the number of training sample is large, the cost $J(\theta)$ may be very expensive to compute in terms of computation and memory. For this reason, the direction of the gradient is computed on a *mini-batch*, a smaller set of the training samples. It is demonstrated in the literature [105] that a gradient descent based on mini-batches is faster and provide a reliable estimation of the cost function computed over the entire training set. Every time the parameters are updated after the computation of the cost function over a mini-batch, it is referred to as *iteration*. The term *epoch* indicates that the entire training set has been processed with smaller mini-batches. Presenting a realistic class distribution in each mini-batch is highly beneficial for the stability of the network [106]. For ASC, the samples in the mini-batch correspond to spectrogram segments with their derivatives. These shorter segments are then randomly shuffled before starting the learning process so that highly correlated segments (i. e. consecutive segments picked from the same recording) are unlikely part of the same mini-batch. Thus, the parameters update is computed from samples which are highly uncorrelated and coming from different classes. This improves the representative power within the mini-batch thereby producing more reliable network parameters.

6.3 OPTIMISING THE CNN

The training of neural networks involves a careful choice of hyperparameters (e. g. type of inputs, number of channels, activation functions, number of layers, etc.).

6.3.1 *Standard practise*

Instead of evaluating all possible combinations of these hyperparameters, standard practice involves finding the best hyperparameters through heuristics. During the selection of the optimal CNN configuration, only a limited set of hyperparameters were fully tested on the DCASE 2016 development set (number of layers, type and number of channels, segment length) whereas other choices were decided *a-priori* based on standard practise. These choices and their reasoning are described as follows:

- the learning rate η refers to the rate of direction of the gradient ∇_θ with respect to the cost function $J(\theta)$

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta J(\theta^{(t)}), \tag{28}$$

where t is the iteration index among T iterations. The selection of the learning rate is a delicate operation since it may influence convergence towards a stable solution. A too small learning rate will slow down convergence whereas a too big value would cause the GD algorithm to iterate arrival in a *saddle* point. In that sense, an adaptive learning rate is artificially created in a range $[\eta_{start}, \eta_{stop}]$ with a step equal to $\frac{\eta_{stop} - \eta_{start}}{T-1}$. Thus, the learning rate decays linearly with the number of epochs. In the experimental works later in this chapter, the range is set to $[\eta_{start} = 0.02, \eta_{stop} = 0.0002]$ with $T = 50$;

- varying the learning over time is not effective when the error surface has gradients in different directions. This produces a phenomenon which makes gradient descent oscillate between the gradients without reaching a local optimum [107]. Momentum is a method which helps to avoid such oscillations and to adjust the gradient toward the most significant direction. It achieves this by adding a fraction γ of the update vector coming from the previous iterations to the current updates [108]. Like an acceleration term, the momentum increases if the past updates have the same direction of the current gradient while it is reduced if the direction changes. However, this method tends to be sub-optimal because the momentum is blindly following the slope. Nesterov accelerated gradient (NAG) partially solves this problem by computing the gradient not only from the current parameters $\theta^{(t)}$, but on the approximate future position of the parameters given by the current parameters $\theta^{(t)}$ and the previous update vector $v^{(t-1)}$

$$\begin{aligned} v^{(t)} &= \gamma v^{(t-1)} + \eta \nabla_\theta J(\theta^{(t)} - \gamma v^{(t-1)}) \\ \theta^{(t+1)} &= \theta^{(t)} - v^{(t)}. \end{aligned} \tag{29}$$

ASC experiments were performed using a value of γ set initially to 0.9 but increased to 0.99 as the number of epochs increase. In this way, the approximate future direction will depend more on the previous updates vector than on the current parameters;

- a rectifier liner unit (ReLU) is adopted as an activation function $f(x) = \max(0, \boldsymbol{w}\boldsymbol{x} + \boldsymbol{b})$. The main advantage of ReLU is to *smooth* the effect of the *vanishing gradient* problem [109], which happens when the gradient of the network's output with respect to θ diminishes exponentially with the layers of the network. Some activation functions (e. g. sigmoid) compress huge parts of the input space into a small range. This problem can be addressed by employing ReLU activation function which, instead, has a range $[0, \infty]$;

- weight initialization determines the starting point of the GD algorithm and highly impacts the training capacity of the network. If initial weights are too high or too low, the setting of the learning rate will be more sensitive. Recent studies by Glorot [110] have proven to be effective by initializing weights from a distribution to zero mean and variance $= \frac{2}{n_{in} + n_{out}}$, where n_{in} and n_{out} are the number of neurons in the first and last layers of the network respectively. In the experimental works presented in this chapter, this weights initialisation procedure is preferred.

6.3.2 *Regularisation techniques*

The way we present the data to the CNN determines its learning capacity. Normalising input data to zero mean and unit standard deviation is a common pre-processing step applied at the initial stage of the network. Instead of computing the normalisation to the input data, it is applied to each mini-batch data after layer transformation. In this way, the outputs of each layer are normalised avoiding the so called *internal covariate shift* effect [111]: when they pass through a deep architecture, data progressively loose their normalization resulting in too big or too small values. This has an impact on the deeper layers, which have to adjust to this change thereby slowing down learning. Batch normalization has the advantage of reducing the learning time while improving classification accuracy.

Parallel to batch normalisation, regularization techniques avoid overfitting when using relatively small datasets. One of the most popular technique is so called *dropout*, which consists in the dropping out of hidden units according to a certain probability. At each training iteration, a random subset of hidden units is temporary disabled by multiplying the input to these units by 0. This forces the network to find robust features that do not depend on the presence of particular neurons [112]. *Dropout* forces each neuron to create the best representation of the data at the individual level.

6.3.3 *Hyperparameter selection*

This section describes the selection process of some of the network hyperparameters. These hyperparameters were selected based on the performance obtained on the DCASE 2016 development set. The entire process is reported in Tab. 9.

The selection follows the same order of Tab. 9 so that the selection of a specific hyperparameter (e. g. the number of layers) depends on the previous hyperparameter choice (e. g. the type of regularisation method). Starting from the sub-table (a), the first choice concerns the type of the regularisation method. Results from single channel log-mel spectrograms as input shows that the combination of dropout and batch normalisation increases performance. Sub-table (b) confirms that 2 convolutional layers are better than 3 layers.

As shown in sub-table (c), the number and type of channels influences the entire network performance. An accuracy of 78.5% is achieved with a 2-channels configuration, where the first channel is the log-mel spectrogram and the second is the time derivative (Δ). The addition of second order derivatives ($\Delta\Delta$) as third channel does not improve further the performance. Interestingly, the use of stereo information is better than using a single channel (from 76.5 to 77.5) but is worse than that achieved with the log-mel + Δ configuration.

A Δ window shift (τ) of 4 (so that the length of the window is 9, $\pm$ 4 the considered frame) gives even better performance (sub-table (d)). Finally, the segment length from which the log-mel and derivatives are computed is tested. The optimal length for the 2-channel input is 1.2 seconds (sub-table (e)).

6.4 EXPERIMENTAL RESULTS

Described here are specific details of the proposed CNN implementation and ASC results for the DCASE 2016 database [26].

6.4.1 *Database & protocols*

The dataset used for evaluating the proposed CNN is the DCASE 2016 database which consists of 15 acoustic scenes. All the different contexts were recorded by a binaural microphone

Method	**dropout**	**batch normalisation**	**dropout + batch normalisation**
1 channel (log-mel)	73.9%	75.6%	76.5%

(a)

Method	**2 conv layers**	**3 conv layers**
batch normalisation + dropout + 1 channel	76.5%	75.6%

(b)

Method	**log-mel + Δ**	**left mic + rigth mic**	**mic average + difference**	**log-mel + Δ + $\Delta\Delta$**
batch normalisation + dropout + 2 layers	**78.5**%	77.2%	77.5%	68.5%

(c)

Method	$\tau = 2$	$\tau = 4$	$\tau = 8$
batch normalisation + dropout + 2 layers + 2 channels	72.7%	**78.5**%	64.7%

(d)

Method	**0.6s**	**1.2s**	**2.5s**
batch normalisation + dropout + 2 layers + 2 channels	67.8%	**78.5**%	75.2%

(e)

Table 9: The hyperparameters selection, based on performance of DCASE 2016 development set. Several hyperparameters are tested and selected according to the highest accuracy. The selection follows this order: (a) the regularisation methods; (b) the number of convolutional layers; (c) the number and types of channels; (d) the window shift τ of the first derivatives Δ; (e) the segment duration.

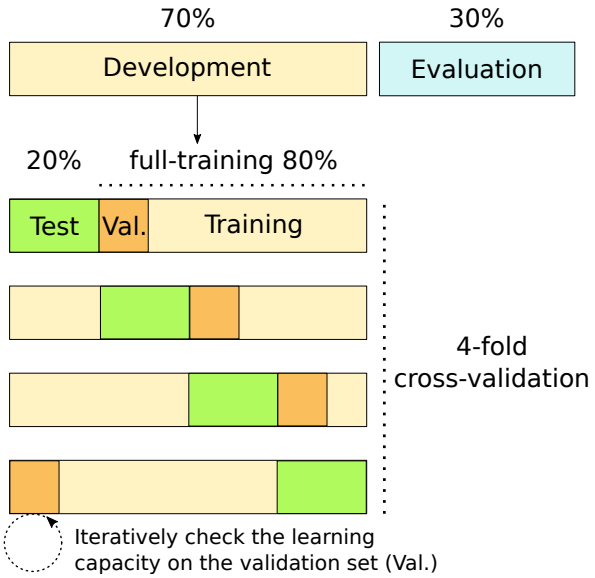


Figure 38: The DCASE 2016 challenge protocol. The database partitioning into development and evaluation set.

using 44.1 kHz sampling rate and 24 bit resolution. Each recording consists of 30s of audio signal. The stereo sounds are very similar to the ones reaching a human hearing system. Here are the scene labels present in the DCASE 21016 database: *bus, cafe/restaurant, car, city centre, forest path, grocery store, home, lakeside beach, library, metro-station, office, residential area, train, tram, urban park.*

The challenge is composed by two subsets: development and evaluation. Fig. 38 depicts the complete protocol. For each acoustic scene, 78 segments (39 minutes of audio) are part of the development set, while 26 segments (13 minutes) are kept for the evaluation. The total amount for the development set is 9h 45min and for the evaluation set is 3h 15 minutes. A 4-fold cross-validation setup is provided for the development set in order to make results reported with this dataset uniform. The cross validation is done at recording level, not frame level. In the experimental results reported in this chapter, the DCASE 2016 development set results refer to the results on the test set (see Fig. 38).

In addition to the standard training-test split, a portion of the original training set is used to iteratively control the generalisation capability during the learning phase of the CNN. This has the advantage to derive the optimal parameters after few epochs. This process has two phases:

1. in the first phase, the original full-training set is split into two separate sets. The training set is used to learn the network parameters while the validation set serves as a *controller* of the learning capacity;

2. in the second phase, once the learning capacity has been validated, the CNN employs the full-training data (80% of the development set).

Concerning the baseline detailed in [26], the parameters for MFCC-GMM baseline system are listed as follows:

- MFCCs frame-size: 40ms (50% hop size);
- number of Gaussians for each class model: 16 components;
- feature vector: 20 MFCC coefficients (including C0) + 20 Δ + 20 $\Delta\Delta$.

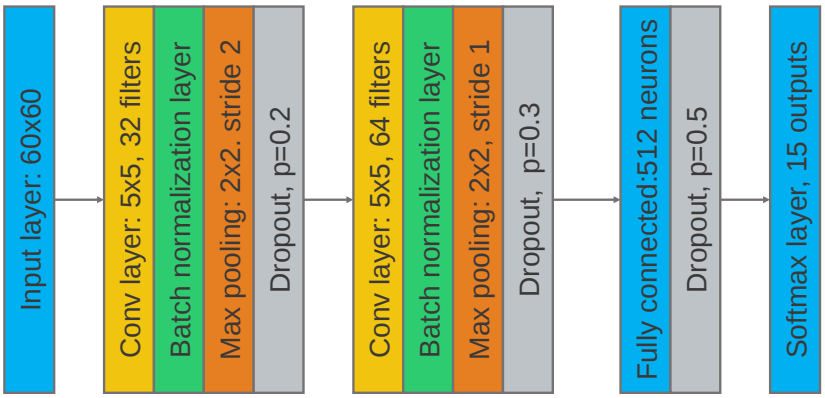


Figure 39: The global architecture of the proposed CNN with the implementation details of each layer.

6.4.2 *Implementation details*

This hyperparameter selection determines the *candidate* system which is then evaluated on the DCASE 2016 evaluation set. This system is referred to as `Battaglino_2` in order to distinguish it from the first CNN version `Battaglino_1` whose details and implementation can be found in [113]. The preference for the `Battaglino_2` rather than `Battaglino_1` system was dictated by a higher accuracy on the training and test set thereby suggesting a greater generalisation capability.

The following implementation details refers to the `Battaglino_2` system which reached a higher accuracy on the test sets. Audio signals are first treated in the usual way involving the application of the discrete Fourier transform to 40ms frames with an overlap of 20ms. Static spectrograms are formed from magnitude spectra which are passed through a bank of 60 log and Mel-scaled filters with a maximum frequency of 22050 Hz. Dynamic Δ spectrograms are calculated in the standard way with a time-window of 9 frames. Each 30s clip of the DCASE database is thus split into 25 segments of 1.2 seconds duration. Each segment is furthermore represented with both static and dynamic spectrogram segments, as illustrated in Fig. 35, resulting in input data of 60 bands $\times$ 60 frames.

The `Battaglino_2` system with its details is displayed in Fig. 39. The CNN has 2 stacked pairs of convolution and pooling layers. The first convolutional layer contains 32 filters each of which spans 5 frequency bands and 5 frames. On account of the relative dimensions of spectrograms and filters and the overlap inherent to the convolution, the frequency and time resolutions of the filters respect the shape of the input segment. Both pooling layers perform max-pooling over 2 adjacent units in both frequency and time, reducing by one half the dimension of the previous convolutional layer, with a stride equal to 2. A second convolutional layer creates 64 feature maps using filters each spanning 5 bands and 5 frames.

A smaller number of filters in the first layer captures low-level pattern structures. In ASC, these patterns correspond to horizontal/vertical edges, chirps and simple structured patterns. These are then combined to form more complex and high-level features which describe an acoustic scenes. Under this assumption, the number of filters is doubled in the second convolutional layer.

The fully connected layer has 512 neurons and is followed by a *softmax* function which returns output probabilities for all of the 15 DCASE 2016 classes. Regularization is performed using a growing dropout probability, and referred before convolutional and fully connected layers. To preserve the creation of the features in the initial layers, the dropout is set to 0.2 and then gradually increased to 0.3 and 0.5. Data is treated in batches of 100 input samples

Method	DCASE 2016 dev set	DCASE 2016 eval set
GMM (baseline system)	72.6%	77.2%
`Battaglino_2`	**78.5**%	**84.4***%

Table 10: ASC performance for the DCASE 2016 development (dev) and evaluation (eval) sets. The * indicates that CNN results are significantly different from those of GMM at 95% confidence interval according to a Wilcoxon signed rank.

and the network is trained for 50 epochs. The learning rate is linearly chosen in the range $[0.02, 0.0002]$ with an initial momentum of 0.9, which is increased linearly to 0.99 for the final epoch.

6.4.3 *DCASE 2016 results*

Classification results are illustrated in Tab. 10 for both the Gaussian mixture model baseline system [26] and the CNN approach (`Battaglino_2`). Results show average classification accuracy over 15 classes for both development and evaluation sets: for the DCASE 2016 development set, the average accuracy is seen to improve from 72.6% for the baseline system to 78.5%; for the DCASE 2016 evaluation set, the accuracy passes from 77.2% to 84.4%.

Results clearly show the benefit of using a CNN architecture for solving an ASC problem. The baseline GMM system is outperformed on development and evaluation set. The first submitted system `Battaglino_1` achieved an accuracy of 80% using a 2 convolutional layered CNN architecture without batch normalisation. The second system `Battaglino_2` adopts the batch normalisation and a squared 5×5 filter shape. The proposed deep architecture is still able to outperform a standard MFCC-GMM system through the automatic learning of meaningful features.

6.5 QUALITATIVE EVALUATION OF THE CNN ARCHITECTURE

While competing with most standard systems, CNNs can be used as a *black box* without any insights into what it is happening inside the network. In particular, experimental works in Chapter 5 about LBPs gave importance to time-frequency patterns which represents a global scene with its composing local features. CNN architecture seems to share with LBP the same concept of *locality* even if is represented with a concatenation of convolutional layers. *What type of automatic features are extracted from the data and how are they relevant to ASC?* still remain unsolved questions.

Therefore, a complete evaluation should involve both performance and qualitative analysis. The qualitative analysis includes the representation of the convolutional filters, the feature maps and the intermediate data transformation after each network layer.

6.5.1 *Filters and feature maps*

The 32 convolutional filters of the `Battaglino_2` system are learned in the first convolutional layer. These filters refer to the first channel (log-mel) of the input data and are illustrated in Fig. 40 (a)). The CNN learns a set of time-frequency filters: some of them better represent vertical lines (e. g. filter numbers 7 and 31); others reflect stationary patterns (e. g. filter numbers 6, 20 and 32). Note that in Fig. 40 (a), black *pixels* represent 0 values (non-active) while white 1 values (active). These filters perform a similar role to the LBP uniform patterns in Fig. 29, with the main difference being that they are extracted directly from the data.

In each convolutional layer, a *feature map* is the result of a filter applied to the receptive field of the previous layer. A given filter is applied across the entire input, with a stride of 1 pixel. Each time the filter is applied to a new pixel position, an activation of the corresponding neuron is collected and drawn in the feature map. This process is applied in the same way for the other 32 filters. In order to show this process and see if some filters are more active depending on the type of signal, the same *toy-problem* recordings used in LBP analysis (5.2.4) are used here: a series of impulsive clicks, a sinusoidal tone at 1kHz and white noise. The filters and activation outputs of the first convolutional layer for each input signal are displayed in Fig. 40 (b), (c), (d).

With respect to the input signals, some insights can be derived about which type of filters are active. For the *click* signal, filters with a vertical component (7, 31) activate neurons in presence of impulsive sounds. Concerning the stationary tone, other feature maps (6, 20, 32) seem to better capture the temporally stationary tone. For the white noise, most of the activation outputs are non-active (black) with some specific feature maps (3, 14, 24, 26) being activated.

6.5.2 *Fully connected layer*

The fully connected layer is an essential component of the CNN architecture. It represents the segments as a combination of local feature maps. Neurons in this layer are connected to the high level features created by the previous convolutional layer, determining which feature maps are the most representative of the acoustic scene. In other words, the fully connected layer has higher activation values for the most discriminative feature maps.

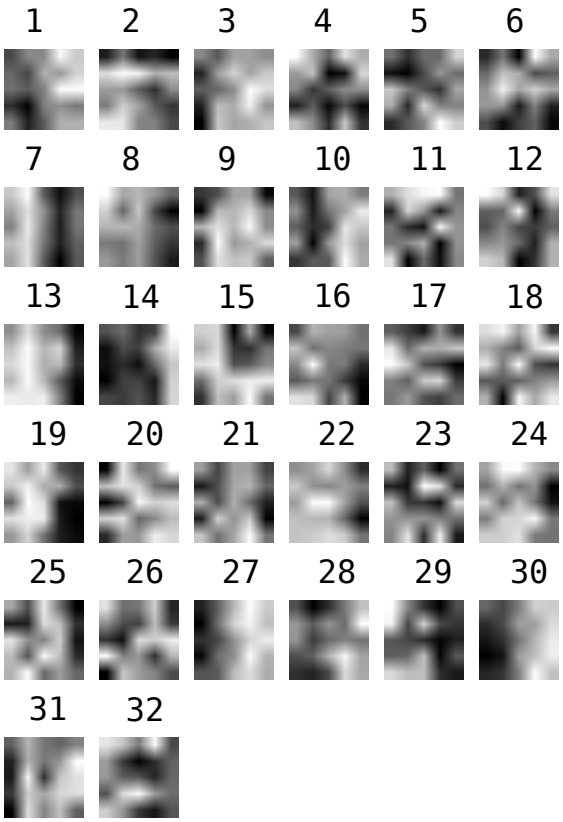
Depending on the nature of the acoustic scene, the activation distribution may change. As for the activation outputs of the fully connected layer, also the LBP algorithm reflects the same idea of describing a global scene with a histogram of local patterns. The activation outputs of the fully connected layer could be used as a feature vector for standard classifiers (e. g. SVM) in the same fashion as LBP histograms [114].

6.5.3 *t-SNE for CNN*

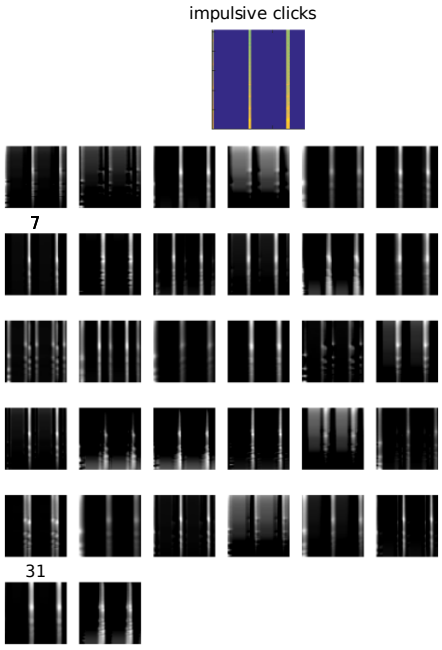
Local patterns are captured through a combination of convolutional and pooling layers which produce significantly higher level representations of the input data after each layer.

The perspectives in [115] provides a rather intuitive and plausible explanation of this multi-layered representation. The intuition behind the work in [115] is that complex data (such as images, audio, text) lie intrinsically in a non-linear manifold space. A manifold is a space characterized by being only locally Euclidean, where local refers to the n^{th} neighbours close to each point. This perspective supposes the existence of manifolds of lower dimensions compared to the original data. Let analyse a complex scene with spectrogram-based features. This space may have thousand of dimensions, but probably only a subset of them are relevant or necessary to the class. If there was a way of automatically finding the most appropriate manifolds of complex spaces, it would project the data in a lower dimensional space, such that the structure of the original space is preserved. CNNs perform this space transformation by *unfolding* complex regions through the convolutional operations.

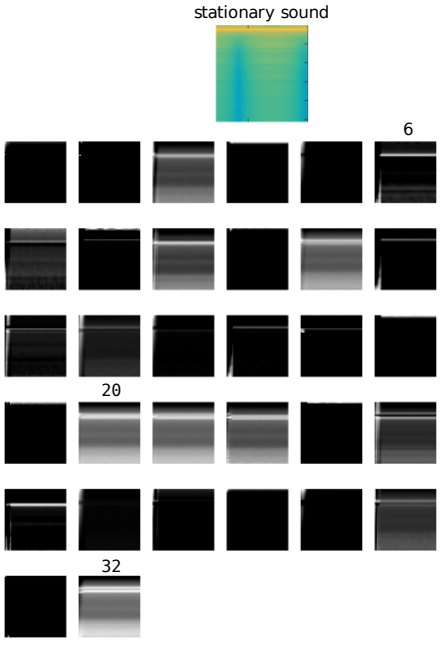
The key idea is that the intermediate hidden layers extract the underlying structure from the original data. The deeper an architecture, the easier to unfold and flatten a complex non-linear data space. The resulting features are then more suited to linear separation. Interestingly, the unfold of *manifolds* form the basis of the t-SNE visualisation, the non-linear dimensionality reduction technique presented in Chapter 4. In fact, t-SNE describes the pattern of the original data on a manifold by representing sample pairwise distances.



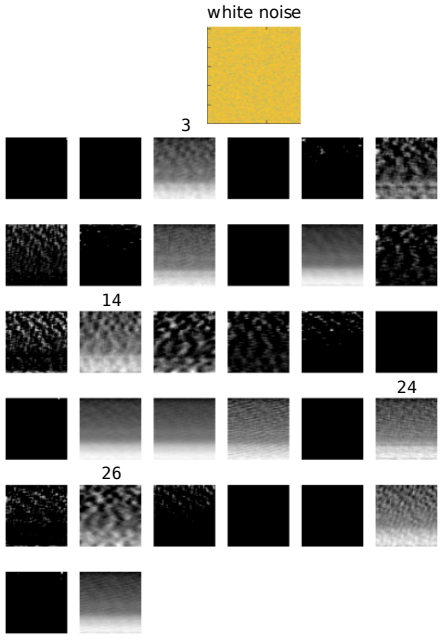
(a)



(b)



(c)



(d)

Figure 40: The insights of the first layer of a convolutional layer: (a) the 32 filters of size 5×5; (b) the activation output of the series of impulsive clicks; (b) the activation output of the tone at 1kHz; (d) the activation output of the white noise. Note that the white colour indicates that the corresponding neuron is active while the black colour expresses a non-active neuron.

t-SNE has been demonstrated to outperform linear dimensionality reduction (e. g. PCA), especially in the case of complex data. In order to verify the manifold perspective, t-SNE is applied to the outputs of convolutional and dense layers. A randomly selection of 20% of DCASE 2016 evaluation data is used, while respecting the same proportion for the 15 classes. As for the experiments reported in Sec. 4.1, a PCA transformation is then applied to reduce dimensionality while retaining 98% of the global variance. Perplexity is set to 100 and the trade-off $\theta = 0.9$, due to the huge number of samples to be processed.

Fig. 41 presents the t-SNE visualizations applied on the data transformed after the CNN layers. The CNN architecture follows the same configuration of the `Battaglino_2`. The distribution of the different classes across the layers changes as the network becomes deeper. In fact, each new layer takes advantages of the data transformation of the previous layer. This transformation is visible in Fig. 41: the separability increases as the architecture becomes deeper. Another observation is that the second convolutional layer seems to have the biggest impact on the transformation, whereas the fully connected layer seems to not be showing such significant improvements. Each of the 15 classes in Fig. 41 are indicated with a different colour and marker, even though t-SNE is based over sample pairwise distances with no knowledge about the class labelling. t-SNE visualizations show that, with the second convolutional layer, samples belonging to the same class tend to be clustered together with a smaller overlap between different classes.

6.6 CONCLUSIONS

This chapter describes a promising application of CNNs to scene classification. In contrast to past works which used almost exclusively hand-crafted features, the work presented in this chapter shows how CNNs can be used to automatically learn local patterns from spectro-temporal representations. Results on a public, standard dataset such as DCASE 2016 confirms the validity of the proposed approach, reaching 84.4% of accuracy on the evaluation set. The CNN approach outperforms the MFCC-GMM baseline by 7%.

Hyperparameters tuning remains one of the biggest limitations. Hyperparameters to be tuned concern different aspects of the network: from the architecture (topology, number and nature of layers, size and shape of the convolutional filters, activation functions), to the type of input (raw audio, spectrogram, filter-based spectrograms) to the training procedure (learning rate, momentum). Due to the training time, an exhaustive evaluation would be infeasible. Only the most significant hyperparameters were fully tested. Based on this tuning, the best system obtained has 2-layers, 2-channels (log-mel + Δ) and segments of 1.2s. Deep learning approaches offer one solution to the limitations of hand-crafted features for ASC. Local patterns are captured through a combination of convolutional and pooling layers which produce higher level representations of the input data. Another pillar of this chapter is the insights into what the network is learning from the data. Too often, deep learning research is performed blindly in *black box* fashion. It is therefore argued that, while based on experimental intuitions, what the networks learn and how they distinguish different acoustic classes is as important as the results they provide.

The visualisation of convolutional filters and feature maps suggests there exist similarities between LBP and CNN, in terms of local patterns. In contrast, the local patterns are pre-defined in the LBP approach (i. e. uniform patterns) whereas CNN extracts patterns directly from the spectro-temporal data. As the architecture becomes deeper, each hidden layer will produce a higher-level representation of the input data. The benefit of this data transformation is then visible using t-SNE, a dimensionality reduction technique which visualizes the intermediate representation of the data passing through the network. Results of this visualization seem to confirm a *manifold* perspective: in other words, the operations

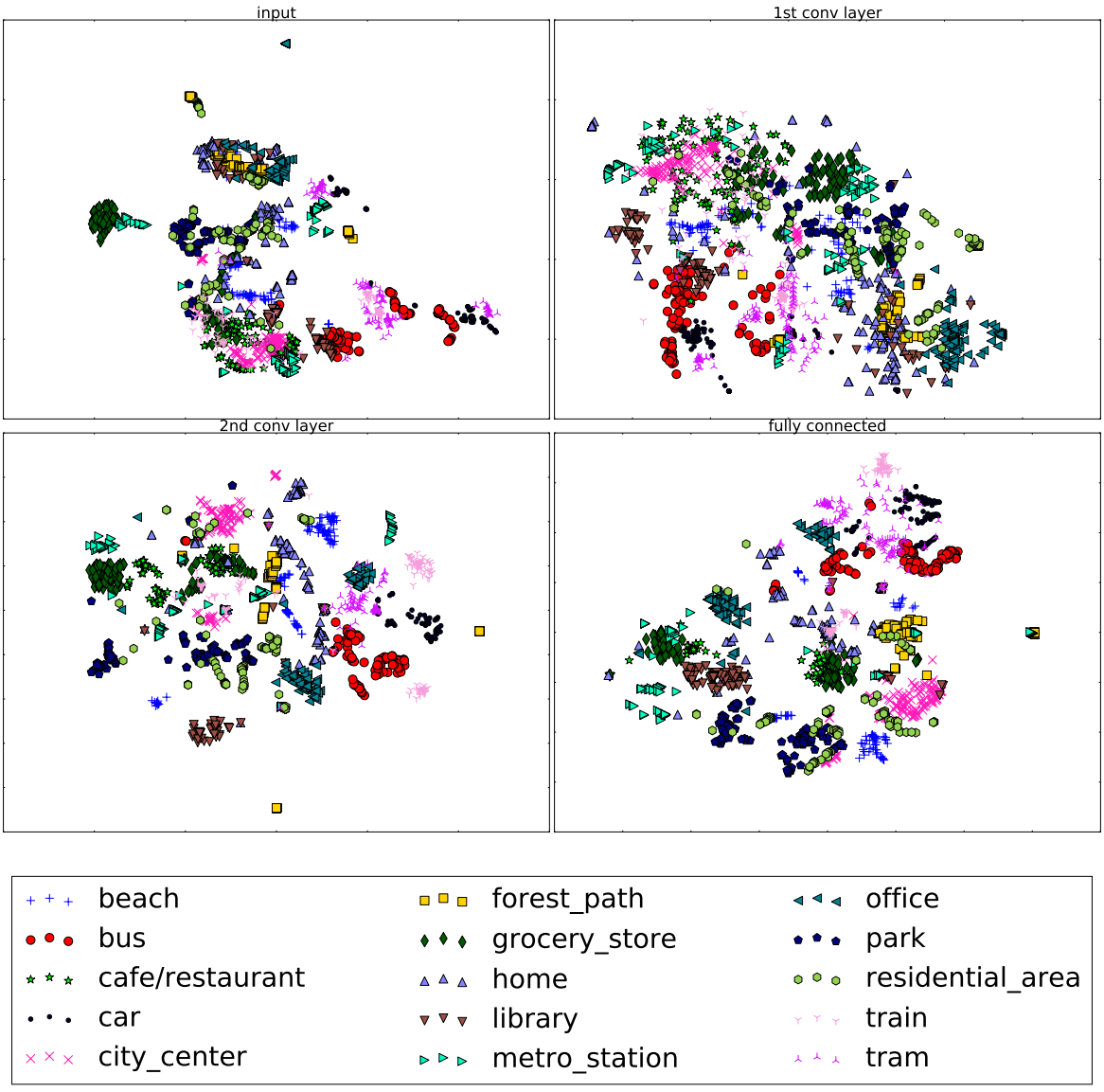


Figure 41: t-SNE visualization of the intermediate outputs of a 3-layered CNN.

performed at each layer flatten and linearise non-linear areas of the original space so as to simplify classification.

7

DCASE 2016 CHALLENGE

This chapter concludes the fundamental research (Part 1) with a review of the most significant systems submitted to the DCASE 2016 challenge. The chapter presents results and the most popular trends in ASC domain. As for the previous 2013 edition, the DCASE 2016 [26] advanced the development of ASC and provided an occasion for companies and universities to discuss about future research directions. The large number of participants to DCASE 2016 challenge demonstrated an increasing interest for new topics such as ASC and AED. There were 82 submissions (3 times more than the previous DCASE 2013 challenge) with ASC alone having 49 submissions. More companies involved themselves in the exploration of this topic (Google, Audio analytics, Soundintelligence, Huawei, NXP, Microsoft, Franhaoufer IDMT) together with a growing community of universities and research laboratories.

The structure of this chapter is organised as follows: Sec. 7.1 summarises the main trends of the DCASE 2016 challenge in terms of features and classifiers methods; a specific review of the most effective methods is presented in Sec. 7.2; Sec. 7.3 concludes the review of the DCASE 2016 submissions .

7.1 TECHNOLOGICAL TRENDS

In order to visualise the main ASC trends, the number of submissions per type of classifier and feature is summarised in Fig. 42. *Technological trends* are defined as methods which then become the standard in a particular domain. Examples in audio-related fields involve the adoption of deep learning techniques in automatic speech recognition (ASR) or music information retrieval (MIR) tasks. Even though not correlated to evaluation performance, charts in Fig. 42 provide a global overview of the most popular methods. Unsurprisingly, deep learning techniques are employed by almost 40% of submissions. The reason of this adoption is therefore related to the automatic creation of features from spectrogram-like data (mel energy, raw spectrogram, CQT). MFCCs remain the standard feature extraction method for one third of the submissions. Ensemble methods (i. e. the fusion of scores or class predictions coming from a combination of different systems) show the most promising results. In fact 3 out of the 5 most performing systems combine MFCC-based methods with deep learning approaches.

7.2 SUBMISSION REVIEWS

In the following section a more detailed review of the best methods is presented. The submission names are the same used in DCASE 2016 results in Fig. 43. The first proposed system `Battaglino_1` achieved an accuracy of 80% using a two convolutional layers without batch normalization. The second system `Battaglino_2` adopts the batch normalisation and a squared 5×5 filter shape. With an accuracy of 5% less than that of the best system (89.7%), the proposed deep architecture is still able to outperform a standard MFCC-GMM system through the automatic learning of meaningful features.

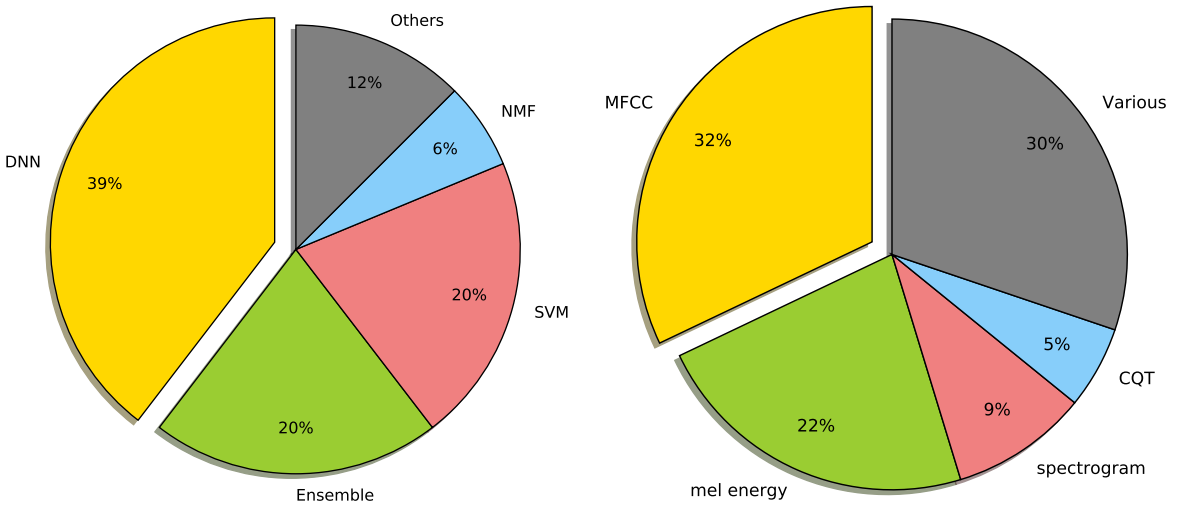


Figure 42: DCASE 2016: the percentages of the most used methods for (a) classifiers and for (b) features.

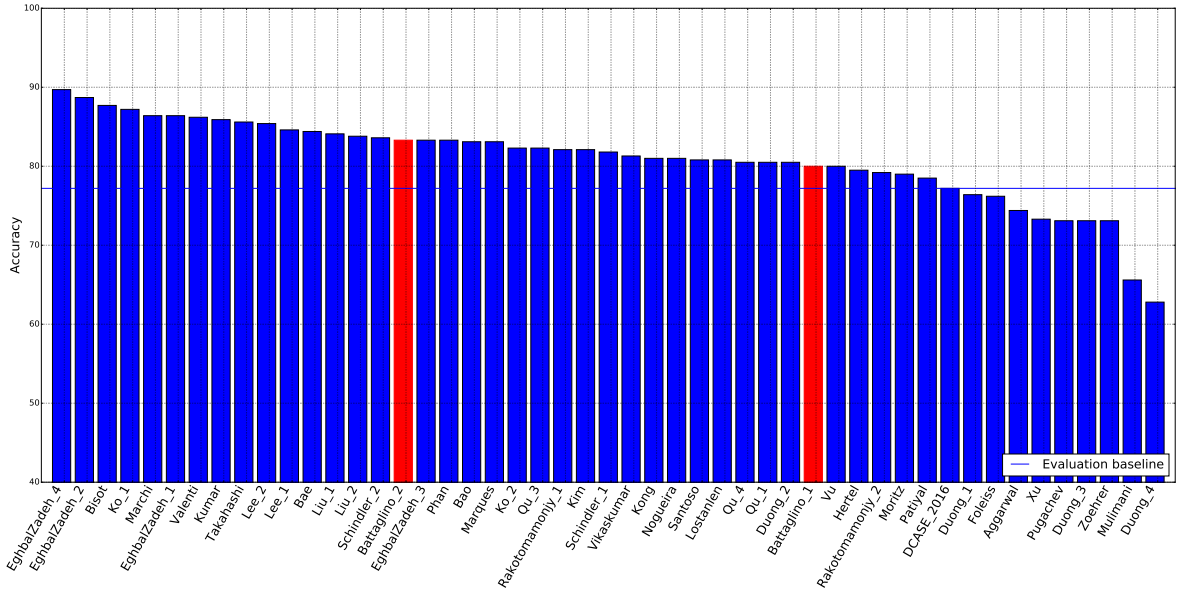


Figure 43: Results on the DCASE 2016 evaluation set. The baseline system has a global accuracy of 77.2% and it is indicated with a solid blue line. The system name follows the same naming of the challenge submissions. In solid red, the proposed CNN-based systems `Battaglino_1` and `Battaglino_2`.

The first two systems Eghbal-Zadeh_4 and Eghbal-Zadeh_2 [116] use binaural information represented by 2 stereo channels (left and right) and their average-difference. First, MFCCs are extracted from each of the 4 channels. Second, starting from channel-based MFCCs, an i-vector is created. The final feature vector is composed of the concatenation of 4 i-vectors. This system alone achieves 88.7%, only 1% less than the same system which combines i-vector and CNN predictions. This suggests that:

1. spatial information, seldom investigated in the ASC literature, is the real differentiator between the top performing systems;

2. the CNN in [116] utilised as inputs single channel spectrograms. As reported in Chapter 6, the CNNs can process multi-channel inputs (i. e. spectrograms extracted from the left and right stereo signal). The level of performance obtained on the test set (Tab. 9) confirms that the use of multi-channel inputs is beneficial to ASC. It is therefore likely that a CNN which employs the same 4 channels configuration, as described in [116], would reach a competitive level of performance.

Many works in the ASC literature express a global scene with its local elementary blocks. The work presented in Bisot [117] is consistent to this *bottom-up* approach by achieving an accuracy of 87.7%: time-frequency representations are decomposed with a non-negative matrix factorisation (NMF), producing a common dictionary of elementary bases. Projections to this dictionary are then used as features for classification.

The 7th best performing system of Valenti [118] is based on CNNs, similar to the systems presented in Chapter 6. Together with this work are other 8 submissions which applied CNNs to DCASE 2016 data (Lee_2[119], Lee_1[119], Bae [120], Eghbal-Zadeh_3 [116], Phan [121], Schindler_2 [122], Schindler_1 [122], Santoso [123], Hertel [124]). Among other deep learning architectures, CNNs provided the best performance with the Valenti implementation (accuracy of 86.2%). While not differing significantly from other CNN approaches, Valenti's work [118] does have some key differences compared to other CNN-based systems: i) a batch normalization procedure is applied to each layer outputs; ii) the network is learned from the entire development set without doing a split into training (80%) and test set (20%). This larger amount of data makes the difference in terms of performance. The main contribution of the proposed CNN (Battaglino_2) with respect to this CNN can be found in the employment of the 2-channel inputs and a shorter segment duration (1.2s).

As for Valenti and Battaglino_2, a popular choice for the input to CNN solution is the log-mel power spectrogram. The approach reported by Schindler_2 [122] uses an alternative CQT transformation to obtain time-frequency inputs to the CNN. Similar to the histogram of gradients (HOG) approach [33], the adoption of the CQT delivers promising results for the ASC task. Not only the type of inputs influences the performance, but also the neural network topology. In Bae's work [120], for example, a parallel architecture combines the sequential information and the spectro-temporal correlation by using a recurrent neural network (RNN) and a CNN respectively. The final layer connects the outputs of the two parallel networks. Xu [125] proposes to integrate the hierarchical taxonomy of the acoustic scenes directly into the deep learning architecture. The network is first trained to classify high-level concepts (e. g. vehicle, outdoor, indoor) and then the specific acoustic scene (e. g. car, park, home).

The findings from the DCASE 2016 challenge confirmed the adoption of deep learning techniques as a competing method for ASC. The flexibility of deep architectures allows researchers to use the same techniques to solve different problems: examples comprise hierarchical taxonomy, temporal recurrence, spectro-temporal locality and stereo microphone input signals. All these findings which may come from other techniques, can be integrated into a single deep learning architecture.

7.3 CONCLUSIONS AND NEXT RESEARCH AXES

This section concludes Part 1 of this manuscript, which reported the main contributions with respect to the fundamental research. In particular, the analysis of the DCASE 2016 submissions complements the literature review of Chapter 2. The chapters structure of Part 1 reflects the chronological order of the contributions proposed in the thesis, from the discussion on the MFCC-SVM system (the DCASE 2013 challenge) until the CNN approach in 2016 (the DCASE 2016 challenge). The focus of Part 1 concerns the research of new features which could better capture the peculiarities of acoustic scenes. Spectro-temporal patterns are demonstrated to be suited to ASC by providing state-of-the-art performance. This includes the employment of local binary patterns (LBPs) and the application of CNNs to time-frequency spectrograms. As shown in Sec. 7.1 and 7.2, CNNs are in fact the most popular deep learning techniques. They also report the higher levels of performance with respect to other popular deep learning architectures such as multi layer perceptrons (MLPs) or recurrent neural networks (RNNs).

Nevertheless, the contents of Part 1 presents only a fraction of the contributions related to this thesis. In fact, the nature of this PhD is industrial: the company sponsoring this research (NXP semiconductors) was interested into the applicability of ASC to real products (e. g. smartphones, earphones). Several aspects of the ASC implementation should be considered when adapting fundamental research to "real-world" applications: the limited memory and computational power; the use of single microphone with a low audio quality or the *always-listening* mode. This *applied* research is described in Part 2. These aspects open new opportunities to enlarge the *spectrum* of research in the ASC domain.

Part II

APPLIED RESEARCH

8

ASC FOR EMBEDDED DEVICES

Part 2 of this thesis relates to the practical limitations of ASC and solutions to them. Except for a few exceptions [29, 126], it is argued here that current ASC solutions do not take into account the need for ASC services running on low-power *embedded* devices. We define *embedded* a device with limited power, memory and connectivity which does not require any connection to an external system to perform classification. Besides ASC computational efficiency, other aspects should be considered in order that ASC be implemented on real consumer products. These aspects are listed as follows:

1. **real-time** (Sec. 8.1). Numerous state-of-the-art algorithms assume access to the entire recording or segment from which features are extracted. This supposition is unrealistic in the case of an ASC system which is required to analyse an audio stream in continuous fashion. Current ASC systems take 30s to provide predictions [33, 56]. These systems may fit a particular type of applications which do not require fast predictions. Other applications, instead, require a faster response. Consider an ASC-based system which applies an adaptive filtering and self-adjusts the quality on a telephone call according to the current scene. Given the mobility of the device, such that the acoustic scene can change during the call, this kind of system should be able to react to a change within few seconds.

2. **low-complexity** (Sec. 8.2). On one side, the training phase is performed *off-line* on external machines. This does not impact the *on-device* complexity. On the other side, *on-device* classification is affected by the model size and the feature complexity; Thus, ASC systems have to consider device limitations, in particular the memory size dedicated for storing the model parameters and the computational complexity of feature extraction.

To address the aforementioned limitations, this chapter reports a real-time and low-power ASC system. Finally, the proposed ASC system is compared with the baseline systems reported in Chapter 3. Experimental results are reported in Sec. 8.3. Conclusions are presented in Sec 8.4.

8.1 REAL-TIME METHODS

A real-time ASC service should extract features in a continuous stream as new audio frames are processed. This chapter reports a real-time implementation of a standard `MFCCs-SVM` system. This standard system is summarised in Fig. 44. There exist two levels of processing: a frame, which has a certain amount of audio samples and acts as the smaller processing-unit from which features are extracted; a segment, which contains a fix number of frames and from which statistics are computed (i. e. mean, standard deviation). While the extraction of feature at frame-level fits the specifications of a low-power device [127], the statistics at segment-level require to store and to process all previous frame-level features.

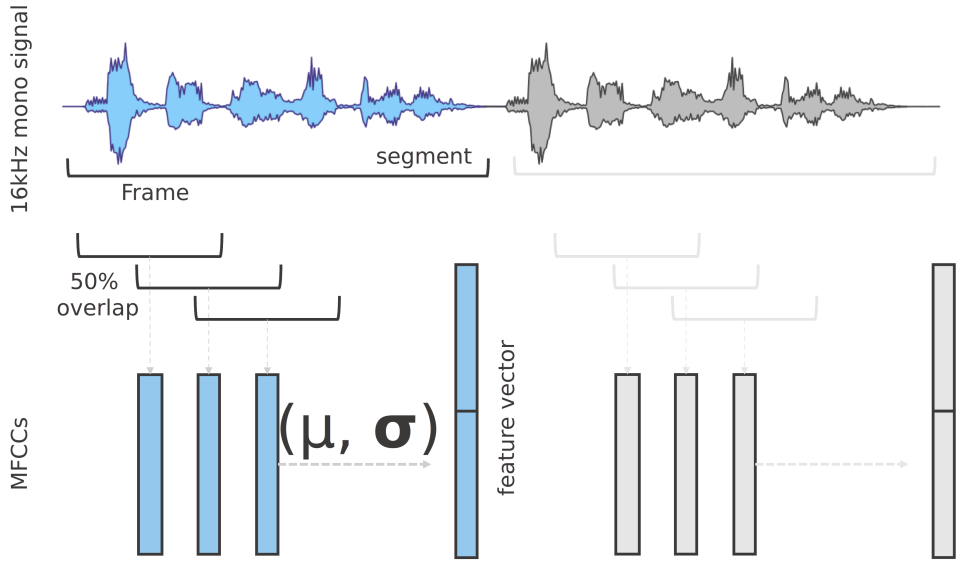


Figure 44: The scheme of real-time feature extraction: MFCCs computed over short frames are integrated over longer segments.

8.1.1 *Recursive estimator*

In a real-time classification, the system analyses one frame at a time and, therefore, statistics over segments have to be estimated using a recursive form. This step involves the estimation of mean and standard deviation statistics through recursive estimators.

Ideally, the estimation of the standard deviation should be performed without storing all previous MFCCs. This corresponds to a gain in processing time (no need to read previously stored values) and memory (no memory needed to store them). Thus, the standard deviation is estimated according to a recursive approach over n frame-wise MFCCs $\boldsymbol{x}_{i=1,\cdots,n}$ [128]. The full method is outlined in Alg. 1.

Algorithm 1 The algorithm for estimating the mean and standard deviation of a continuous stream of $\boldsymbol{x}_{i=1,\cdots,n}$ features

procedure MEAN AND STANDARD DEVIATION RECURSIVE ESTIMATOR($\boldsymbol{x}_{i=1,\cdots,n}$)
  $\hat{\boldsymbol{\mu}}_1 = \boldsymbol{x}_1$
  $\hat{\boldsymbol{\sigma}}_1^2 = 0$
  for $i = 2 \rightarrow n$ **do**
    $\hat{\boldsymbol{\mu}}_i = \hat{\boldsymbol{\mu}}_{i-1} + (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{i-1})/i$
    $\hat{\boldsymbol{\sigma}}_i^2 = \hat{\boldsymbol{\sigma}}_{i-1}^2 + (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_i)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{i-1})$
  $\hat{\boldsymbol{\sigma}} = \sqrt{\hat{\boldsymbol{\sigma}}_i^2/(i-1)}$

The estimated mean $\hat{\boldsymbol{\mu}}_i$ is initialised to the first MFCC in the segment, $\boldsymbol{x}_1$. The estimated variance $\hat{\boldsymbol{\sigma}}_i^2$ is initially set to 0.

From the second frame ($i = 2$), the mean $\hat{\boldsymbol{\mu}}_i$ and $\hat{\boldsymbol{\sigma}}_i^2$ is estimated using the current i^{th} frame with no need to store all previous $(i-1)^{th}$ MFCCs. When i reaches the index of the last frame of the segment n, the standard deviation $\hat{\boldsymbol{\sigma}}$ is computed from $\hat{\boldsymbol{\sigma}}_i^2$. At the end of the iteration, the estimated statistics are used to create the feature vector used by a SVM classifier.

While it estimates mean and standard deviation statistics iteratively, this algorithm suffers from poor adaptation to signal changes. When the iteration index i is too large (e. g. thousands of frames), the incremental term $(\hat{\boldsymbol{\mu}}_{i-1})/i$ tends to 0. This becomes a problem in ASC where changes in the acoustic scene may produce significant variations in the MFCCs.

Due to the need for stable convergence and rapid adaptation, this estimation algorithm poses more than one limitation. In the practical case, it may happen that the convergence time of the estimator will be longer than the actual change in the acoustic scene. An ideal estimator should follow signal changes. These changes correspond to the presence of events or to a scene change. For these reasons, a new version of the recursive estimator is proposed.

8.1.2 *Tandem estimator*

The proposed estimator computes statistics (mean, standard deviation) with two estimators working in *tandem*. Their estimation follows same procedure detailed in Alg. 1, except that one of the two estimators starts after the other with a certain *offset*. This estimation procedure is referred to as a *tandem estimator*. A counter is associated with each estimator and is incrementally updated as a new frame is read. When the counter of one of the estimators is greater than a fixed value max_c, the corresponding estimator is reset to the initial conditions and its counter reset to 1.

Depending on the counter with the larger value, the proposed tandem estimator decides which of the two estimators concurs the most to the final estimation. Once one of the tandem estimators provides its estimation, the corresponding counter is reset. The tandem estimator has the advantage of better following rapid variations in the signal and to be almost insensitive to the number of frames n in the segment.

The complete tandem estimator algorithm is detailed in Algorithm 2. In the initialisation phase of the algorithm, mean and standard deviation estimators are initialised as follows: the global estimator $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}})$, tandem 1 estimator $(\hat{\boldsymbol{\mu}}\mathbf{1}, \hat{\boldsymbol{\sigma}}\mathbf{1})$ and tandem 2 estimator $(\hat{\boldsymbol{\mu}}\mathbf{2}, \hat{\boldsymbol{\sigma}}\mathbf{2})$ are all set to 0, except for tandem 1 mean, which is set to the first MFCC value.

Note that the two counters are independent of the i^{th} frame as they have their internal counters (counter1 and counter2). Every time one of two counters surpasses max_c, the corresponding estimator is re-initialized. The global statistics are then selected depending on the tandem whose counter is the highest.

As an example, Fig. 45 illustrates how the tandem estimator works in estimating the mean of the first MFCC coefficient C0. The choice of C0 is due to the visualisation purpose. In practise, the tandem estimator computes the mean and standard deviation of every multi-dimensional frame-level feature vector.

Segments in this example contain 400 frames. The second estimator (tandem 2) starts at $i = 200$, while the first estimator has not reached the max counter max_c and it is used to estimate the C0 mean. When the second segment starts, the first estimator (tandem 1) is reset and tandem 2 provides the estimation at $i = 400$. This mechanism is repeated for the next segments in a continuous streaming fashion. In Fig. 45, the max counter max_c is set to 600 frames and the offset between the two tandems is equal to 200 frames.

To illustrate how the tandem estimator works in practice, an *ad-hoc* example is provided. Fig. 46 depicts the variation in C0 mean for white noise signal at different energy levels. In this specific example, only the 1-dimensional C0 is considered. For the experimental results reported later in this chapter, the estimation is applied to multi-dimensional mean and the standard deviation. The tandem estimator is then compared with the recursive estimator. As expected, the tandem estimator better adapts to rapid variations in the signal, while the recursive estimator is much slower in converging to the true value.

Algorithm 2 The tandem estimator algorithm of a continuous stream of $x_{i=1,\cdots,n}$ feature vectors

procedure MEAN AND STANDARD DEVIATION TANDEM ESTIMATOR(max_c, offset)
  $\hat{\mu} = 0$
  $\hat{\sigma^2} = 0$
  $\hat{\mu 1}_1 = x_1$
  $\hat{\sigma 1^2}_1 = 0$
  $\hat{\mu 2}_1 = 0$
  $\hat{\sigma 2^2}_1 = 0$
  counter1 = 1; counter2 = offset;
  for $i = 2 \rightarrow$ inf **do**
    $x_i \leftarrow$ compute feature vector of current i^{th} frame
    $\hat{\mu 1}_i = \hat{\mu 1}_{i-1} + (x_i - \hat{\mu 1}_{i-1})/$counter1
    $\hat{\sigma 1^2}_i = \hat{\sigma 1^2}_{i-1} + (x_i - \hat{\mu 1}_i)(x_i - \hat{\mu 1}_{i-1})$
    $\hat{\mu 2}_i = \hat{\mu 2}_{i-1} + (x_i - \hat{\mu 2}_{i-1})/$counter2
    $\hat{\sigma 2^2}_i = \hat{\sigma 2^2}_{i-1} + (x_i - \hat{\mu 2}_i)(x_i - \hat{\mu 2}_{i-1})$
    if counter1 >= max_c **then**
      counter1 = 1
      $\hat{\mu 1}_i = x_i$
      $\hat{\sigma 1^2}_1 = 0$
    else
      counter2 = 1
      $\hat{\mu 2}_i = x_i$
      $\hat{\sigma 2^2}_1 = 0$
    if counter1 > counter2 **then**
      $\hat{\mu} = \hat{\mu 1}_i$
      $\hat{\sigma} = \sqrt{\hat{\sigma 1^2}_i/(\text{counter1} - 1)}$
    else
      $\hat{\mu} = \hat{\mu 2}_i$
      $\hat{\sigma} = \sqrt{\hat{\sigma 2^2}_i/(\text{counter2} - 1)}$

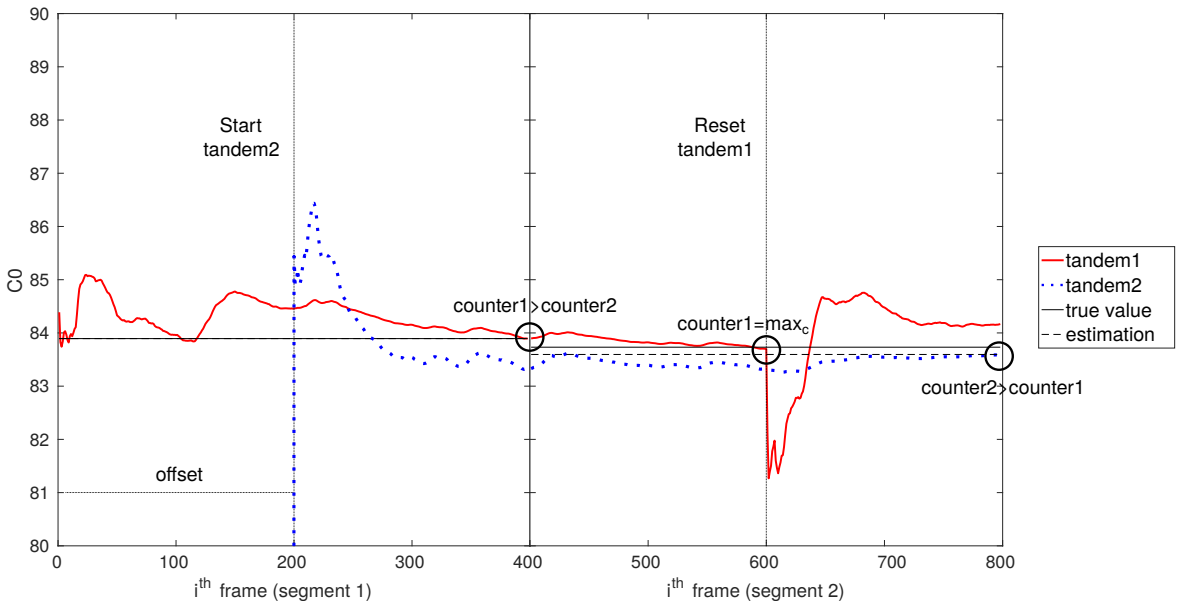


Figure 45: Tandem estimator mechanism on 800 MFCCs frames, split into 2 segments. The offset is set to 200. In solid black the true value of C0 mean, in dashed black the estimation; in red and blue the first and second tandem estimator values as they are computed using the i^{th} frame. Tandem 1 (in solid red line) has reached the maximum counter and it is reset in segment 2.

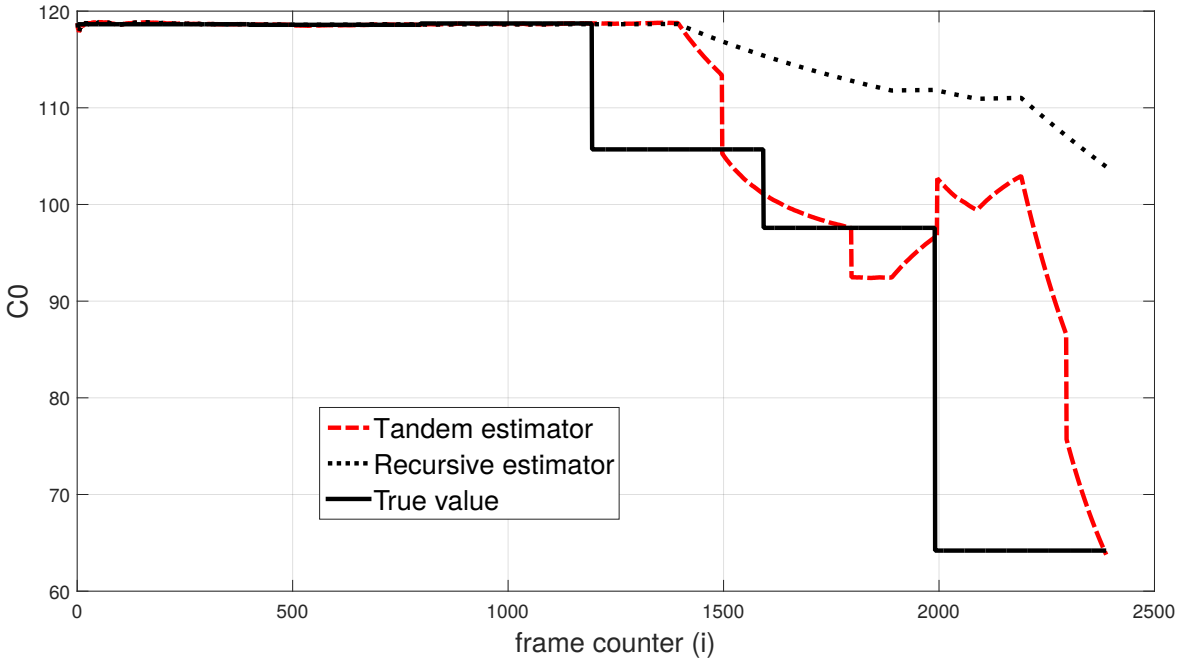


Figure 46: Tandem estimator and recursive estimator adaptation on a varying signal. The mean of C0 in solid black is tracked by the two methods: the tandem estimator (in dashed red line) better adapts to changes; the recursive estimator (in dotted red line) slowly follows the true values.

Nevertheless, the tandem estimator depends on the choice of the max counter and the offset between the two estimators. These external parameters depend on the length of the segments and they have to be manually tuned to minimise the global error between true and estimated values.

8.2 LOW-COMPLEXITY METHODS

Efficiency is especially important with ASC for embedded devices. First, unreliable data connections and the power implications of continually communicating audio data to a remote server make cloud solutions impractical. While running locally on the device itself, computational efficiency is essential to minimise battery consumption. Second, the context is dynamic. The need for *always-listening* ASC calls for algorithmic efficiency. Third, reliable context recognition usually requires context modelling with large amounts of data.

It is argued herein that current ASC approaches are too costly in terms of computation and memory requirements to support *an always-listening* mode. Even though model learning is performed *off-line*, with little or no memory or computation limitations, the testing is performed *on-line* and remains the most critical aspect.

This section describes a reduced complexity ASC system: the principal hypothesis is that feature dimensionality can be highly reduced and that a significant fraction of training samples contains information redundant to the classification. Inspired by related research [129, 130, 131], the proposed method relates to the selection of training samples through clustering and decimation resulting in a smaller number of training samples and therefore a less complex model coming from these data.

8.2.1 *Measures of complexity*

Efficient modelling is thus needed to avoid the processing and storing in memory of large, complex models. As an example, it has been demonstrated in [132] that memory and computational cost of SVM classifiers are proportional to #SVs $\times$ D, where D denotes the feature dimensionality and #SVs the number of support vectors. In the method reported in this chapter the focus has been directed towards the memory improvements but, since also the complexity depends on the same variables, it is argued to be beneficial for both. With large quantities of data being needed for reliable ASC, standard SVM classifiers are typically too complex.

The size of memory has been calculated supposing blocks of 4 Bytes (B) for each SV dimension. As a reference device for an always-listening ASC system is considered a cortex-M4 processor with 80MHz operations per second and 256kB of memory (which should include the code itself). The M4 processor is considered as the reference embedded device for signal processing [1] [133].

Consider 200 kB of available memory for storing the model. Suppose, also, a standard approach with MFCC-based statistics (mean and standard deviation) with D $=$ 26. The maximum number of SVs is: 200kB/ 26 features/ 4 Bytes $=$ 1900 SVs. Generally the memory on a device has to be shared with other software functionalities (e. g. video and audio codec, sensors, speech recognition). This means that the memory available may be much lower than 200kB thereby limiting also the total number of SVs which can be stored in the SVM model.

1 https://www.arm.com/products/processors/cortex-m/cortex-m4-processor.php

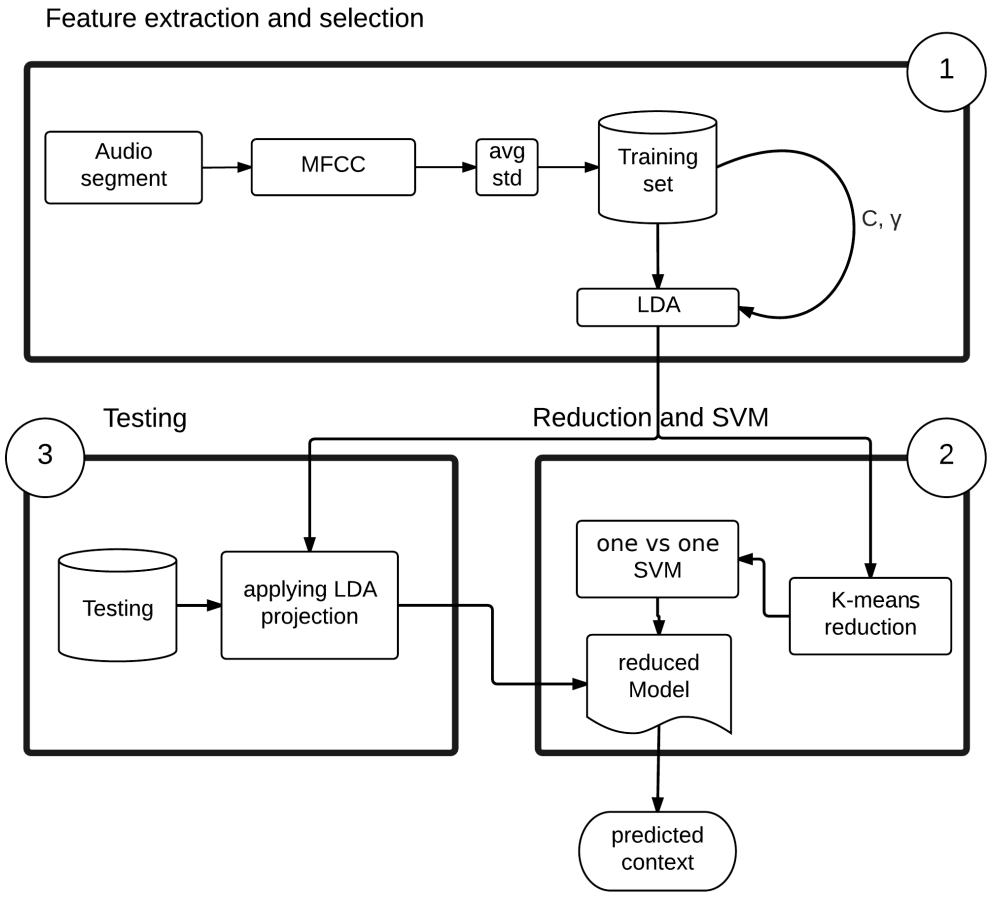


Figure 47: Reduced complexity ASC system: **(1)** feature extraction and selection using LDA; **(2)** SVM training after K-means clustering and corresponding decimation; **(3)** Testing with SVM reduced model, applying the LDA projection.

8.2.2 *Reduced complexity ASC system*

Complexity reduction is achieved through a set of techniques designed to reduce the number of SVs and the feature dimensionality with the common goal of decreasing the memory size and the computational complexity of the testing phase. Before training, feature extraction and selection are performed, followed by reduction of the training dataset. In testing, the feature selection transformation is applied to the test data before classification.

The entire process of complexity reduction is depicted in Fig. 47: before training, feature extraction and selection are performed using linear discriminant analysis (LDA) (1); during training, the SVs of the SVM are learned after the K-means dataset reduction (2); during testing, LDA projection is applied to the test data before the classification.

In the first step, full audio recordings are divided into non-overlapping fixed segments. For each segment, standard MFCCs are computed over short overlapping frames before mean and standard deviation statistics are determined. This produces a single, fixed-length feature vector for each segment.

LDA is then applied in order to reduce dimensionality while improving discrimination. Original training features are projected into a new sub-space where the ratio of *between-class* to *within-class* variability is maximized according to the *Fisher* cost function, as defined in Eq. 18. This problem is treated as a regular eigenvalue problem, where the eigenvectors corresponding to the largest eigenvalues are used to determine a discriminant feature

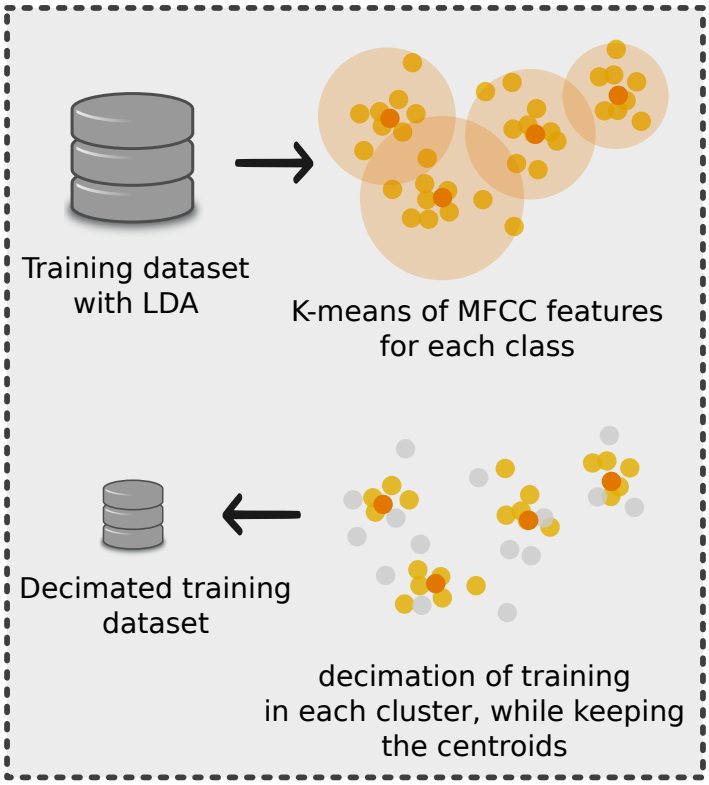


Figure 48: Diagram of data reduction using K-means clustering.

transformation [134]. Since LDA is a supervised technique which utilizes label information from C classes, the maximum dimensionality allowed is equivalent to $C-1$. It is worth mentioning that LDA is applied after a *z-score normalization* on the training set. While maintaining the same level of accuracy of the non-reduced system, LDA reduces the dimensionality of feature vectors.

8.2.3 *Data decimation*

Following the scheme in Fig. 47, step (2) involves the learning of class models from sub-sets $\tilde{\mathcal{X}}$ of original training set $\mathcal{X}$. In the proposed decimation algorithm, the training data is split into the corresponding class samples $\boldsymbol{x}_{i=1,\dots,N_c}$ where N_c indicates the size of samples belonging to the c^{th} class. The data of each class is then clustered into K clusters using a standard K-means algorithm which minimizes the average distance between a set of samples and a set of clusters centres $\boldsymbol{\mu}_{k=1,\dots,K}$ expressed as an objective function:

$$\min \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in \mathcal{X}_k} ||\boldsymbol{x} - \boldsymbol{\mu}_k||^2 \tag{30}$$

where $\boldsymbol{x} \in \mathcal{X}_k$ is the set of samples belonging to the k^{th} cluster and $\boldsymbol{\mu}_k$ is the k^{th} cluster mean. Cluster centroids are initialized randomly. At each iteration, the K-means algorithm attributes samples to their nearest cluster. Cluster means are updated until convergence. The samples attributed to each cluster are then decimated according to uniform selection so that the full distribution is now represented by a subset of the original data $\tilde{\mathcal{X}}$ with the addition of cluster centroids.

Clustering and data selection is performed for each class before a multi-class SVM classifier is trained with the reduced subset. One problem remains in selecting a global

number of clusters K suitable for all acoustic scene samples. Since results seems to depend on the context, the decimation algorithm should be able to automatically estimate a number of clusters for each class data samples [135]. An optimal strategy involves decimation optimised at the class and cluster levels.

8.2.4 *Optimising the number of clusters*

Among other metrics dedicated to evaluate the number of clusters, the silhouette width is a composite criterion which measures the cluster *strength*: in other words how well each sample has been grouped in one of the clusters [136]. The silhouette width is chosen because it can be applied to any distance metric. For each training sample $\boldsymbol{x}_i$ and a given cluster k, the silhouette width is defined as follows:

$$s_i = \frac{b_i - a_i}{\max(b_i, a_i)}, \tag{31}$$

where a_i identifies the average distance of sample i to the other *with-in cluster* samples, b_i is the average distance of sample i to the other set of *between cluster* samples. The average of s_i over all training samples provides a global metric, whose range spans from 0 to 1. Let ASW be the average silhouette width over N samples:

$$ASW = \frac{\sum_{i=1}^{N} s_i}{N}. \tag{32}$$

A high value of *ASW* means that the corresponding K is a reliable estimation of the number of clusters. On the contrary, when ASW values are too low, then the number of cluster K is not reliable. The ASW metric can be applied to the data decimation algorithm as a way of estimating the optimal number of clusters K. Fig. 49 shows the ASW as a function of the number of clusters K, for each acoustic scene for the DCASE 2013 evaluation set. Interestingly, many classes have low ASW values (i. e. below 0.4). This shows that the data has only weak structure. Compared to prior experiments [135] with a fixed value of clusters for all acoustic scenes, learning an optimal K for each class improved performance while respecting the underlying data structure of each acoustic scene.

8.2.5 *A distance-based decimation*

The K-means clustering ensures that the entire feature space is represented after decimation but the decimation is obtained by randomly selecting sample from each cluster and including all centroids. Moreover, the fact of adding cluster centroids after the decimation ensures a consistent representation even at strong levels of decimation.

An alternative strategy involves a distance-based data decimation which aims to improve on the random decimation applied to each cluster. Intuitively, we should remove samples whose contribution to the general data distribution is negligible and retain the most salient.

Inspired by work in [137], the reduction algorithm can be improved by saliency samples according to their distance to the closest cluster centroid. This distance metric is computed using the Euclidean norm of the i^{th} sample $\boldsymbol{x}_i$ of each class set $\mathcal{X}_c$ and the closest centroid $\boldsymbol{\mu}_k$ according to:

$$d_i = \min \|\boldsymbol{x}_i - \boldsymbol{\mu}_{k=1,\cdots,K}\| \quad \forall i = 1, \cdots, N_c. \tag{33}$$

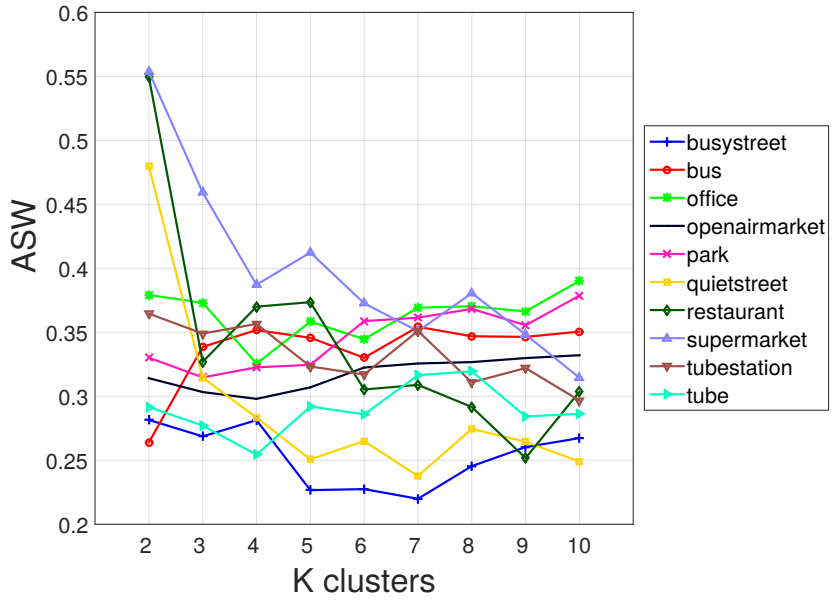


Figure 49: The silhouette values as a function of different K clusters. The highest value for each acoustic scene is then considered as *Optimal K cluster*.

The set $\mathcal{X}_c$ of the c^{th} class is then sorted in descending order based on d_i. Iteratively, the first sample $\boldsymbol{x}_1$ of this ordered list is put into the decimated set $\tilde{\mathcal{X}}_c$. Then, all the samples whose distance is lower than a distance threshold δ are removed from the list (comprising the first element $\boldsymbol{x}_1$). The routine is repeated by filling $\tilde{\mathcal{X}}_c$ until the ordered list is empty. Since samples lower than a distance threshold δ are removed, a higher value of δ indicates a severe decimation.

When the threshold value δ is too large, the decimation may remove all training samples. In this case, only the centroids of each cluster $\boldsymbol{\mu}_k$ are retained. The distance-based decimation method is referred to as *distance decimation* in contrast to the method based on cluster with no distance-based selection, called *data decimation*.

8.3 RESULTS & DISCUSSION

Real-time and low-complexity method results are reported in this section. These are assessed using the DCASE 2013 and the NXP datasets (Chapter 4), employing a standard 5-fold cross-validation partitioning. Averaged accuracy and corresponding confidence intervals are provided for all methods.

8.3.1 *Implementation details*

MFCC features are extracted from frames of 32ms with a 16ms overlap and accumulated to form non-overlapping segments. The frequency range is set to $0-2000$Hz. Mean and standard deviation statistics are extracted over each segment thereby creating a 26-dimensional feature vector. This dimension is reduced to $C-1$ (where C is the number of classes) through LDA projection. Depending on the number of acoustic scenes, the feature dimensionality is reduced to $C-1=9$ for the DCASE 2013 dataset ($C=10$) and to $C-1=4$ for the NXP dataset ($C=5$).

SVM classifiers were implemented with the *LibSVM* library [72], using RBF kernel. For the decimation methods described in Sec. 8.2.3 and 8.2.5, the segment length from which MFCCs statistics are computed is set to 3s with no overlap. Since each audio file is split into 10 segments, a majority vote strategy is adopted to provide a single prediction for each

MFCC-2000Hz	**Standard**	**Real-time**
30s	67% (±8%)	61% (±9%)
10s	70% (±8%)	63% (±10%)
5s	71% (±12%)	59% (±10%)
3s	72% (±4%)	59% (±12%)

Table 11: Standard vs real-time estimation of mean and standard deviation over different segment lengths. Results refer to DCASE 2013 evaluation set.

audio file. This segment length is empirically found to produce the highest performance with the current MFCC configurations (72% of accuracy). Reduction rate is expressed as the ratio of the number of decimated training samples with the original training set size. The reduction rates for the *distance decimation* method are obtained by varying the distance threshold δ between 0.1 and 50; the reduction rate for the *data decimation* is a parameter which determines the reduction for each cluster. The set of training samples which remains after the decimation is expressed with $\tilde{\mathcal{X}}$. In order to compare the different decimation methods, $\tilde{\mathcal{X}}$ size is fix so that every decimation method uses to the same number of samples for learning the model.

8.3.2 *Comparing the tandem estimator with a standard system*

The tandem estimator computes estimation of the feature statistics only for the testing samples. Two methods are then compared: one computing statistics over the entire segment (*standard*), the other adopting the tandem estimator (*real-time*). The tandem estimator is applied to the concatenation of all test recordings representing in this way a single audio stream with no interruption between recordings. This emphasizes the capacity of the proposed method to automatically adapt to context changes (e. g. passing from a *bus* to an *office* scene) reflecting what could happen in practical applications. Segment lengths are 30s, 10s, 5s and 3s with no overlap between segments. ASC results for different segment lengths are presented in Tab. 11. For values below the file duration (i. e. 30s), a majority vote strategy is utilised.

Experimental results in Tab. 11 show a drop in performance between the real-time and standard system. The best real-time estimation corresponds to a 10s segment duration, where both standard and real-time systems achieve a better accuracy. The standard system computed at 3s reports the best performance. Its corresponding real-time system show an accuracy of 59%, with a poor estimation on shorter segments. This is confirmed by results computed over 5s. The fact that in the real-time system the files are all concatenated in a unique audio stream (therefore adapting to scene change) may explain the difference between the two systems.

Although a drop in performance is observed for the real-time estimator, the proposed algorithm for estimating the mean and standard deviation outperforms the recursive estimator and better adapts to rapid changes in the audio stream. In addition, it allows state-of-the-art ASC methods to work in real-time by processing single frame-wise MFCCs.

8.3.3 *Comparing decimation methods*

train set size (% reduction)	SVs	accuracy	memory (KBytes)
800(0%)	462	72%	47
752(6%)	429	71%	15
440(45%)	240	61%	8
232(71%)	135	58%	5
128(84%)	63	61%	2

Table 12: Recognition accuracy, number of support vectors and memory requirements for different amounts of training data reduction, for the DCASE 2013 evaluation dataset. At 0% of reduction, no feature or data decimation is applied.

train set size (% reduction)	SVs	accuracy	memory (KBytes)
162800(0%)	7802	88%	794
6512(69%)	2887	83%	45
1050(95%)	400	81%	6
420(98%)	158	80%	2
210(99%)	49	78%	0.7

Table 13: As for Tab. 12 for the NXP dataset.

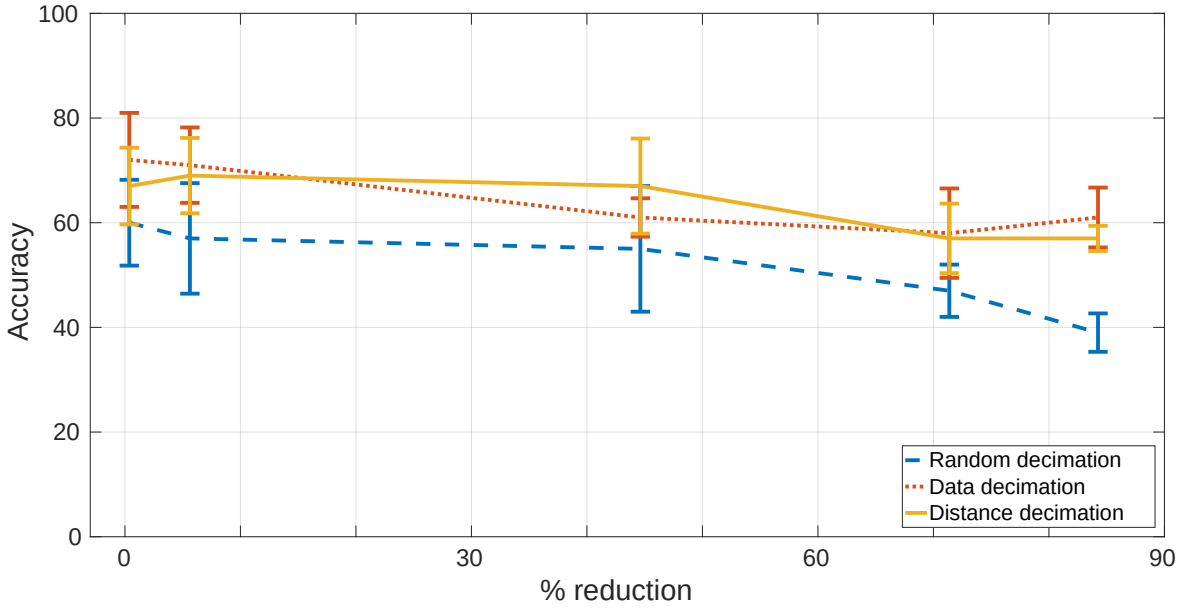
Results for the data decimation method for the DCASE 2013 and NXP datasets are presented in Tab. 12 and 13. The results for the *distance decimation* are similar to those of the *data decimation* method in terms of memory required/number of SVs and therefore not reported here. For the DCASE 2013 evaluation set, at a cost of an accuracy drop of 11%, the amount of SVs is reduced by a factor of 20. Interestingly, for a 6% of data reduction, the difference in terms of accuracy is negligible.

For the larger NXP dataset, results show a significant reduction (99%) in training samples with a drop in accuracy of 10%. Other reductions can be achieved with less severe reduction rates. As an example, a reduction from 794 to 45 kB corresponds to a drop of only 5% in accuracy. In order to demonstrate the benefit of clustering before decimation in terms of memory required, results are also presented for a system which reduces the training data by random data selection without prior clustering. This approach is referred to as *random decimation*. Comparative curves for *data decimation*, *distance decimation* and *random decimation* are reported in Fig. 50 (a) and Fig. 50 (b), for the DCASE 2013 evaluation and NXP datasets respectively. Each curve represents the mean accuracy and confidence interval (CIs) computed over 5-fold cross-validation partitions. Presented in the following are considerations regarding these results:

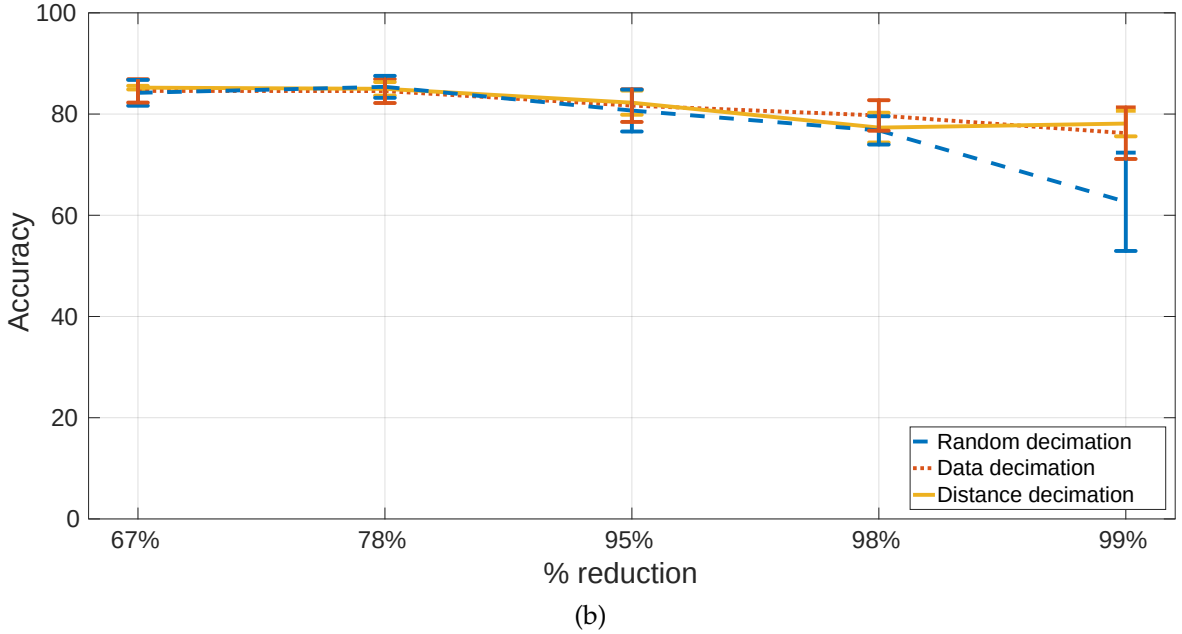
- data decimation and distance decimation outperform random decimation. By varying the reduction rate, one can select a *trade-off* model between complexity and performance degradation;
- accuracy for the DCASE 2013 evaluation set is better for distance decimation at a reduction of 45%. Concerning results for the NXP dataset, data and decimation selection methods follow a similar trend.

8.4 CONCLUSIONS

Although presented separately, the proposed methods (tandem estimator, data decimation, distance decimation) have a common goal, namely the development of a complete real-time and low-power solution to ASC. Hence, a solution of this kind could be implemented in a reference embedded device (e. g. the ARM cortex M4).



(a)



(b)

Figure 50: Data decimation (dotted red line) vs distance decimation (yellow line) vs random decimation (dashed blue line), for (a) DCASE 2013 and (b) NXP datasets.

State-of-the-art methods employ SVMs with MFCC statistics, computed from the entire recording. The proposed real-time approach estimates the mean and standard deviation using recursive estimators working in *tandem*. The tandem estimator better tracks changes in the audio signal and is therefore better suited to real-time ASC. A second limitation concerns the complexity of the statistical model generated during the training phase. The principal idea involves the selective decimation of training data such that a reduced set of support vectors are then required during the testing phase. LDA is applied to reduce the feature dimensionality. K-means clustering is the basis for data decimation, ensuring that the full feature space is adequately represented after decimation. Results on a small, standard dataset and the larger, non-standard NXP dataset confirm the validity of this approach, showing a significant reduction in memory size and computational cost without a severe impact on classification accuracy.

To conclude, this chapter describes the first attempt to build a real-time, low-complexity ASC system for embedded devices. The methods proposed here are not in conflict with state-of-the-art methods whereas providing *sustainable* versions from a low complexity and low memory, real-time point of view. These represent *key* industrial contributions of this PhD.

9

THE OPEN-SET PROBLEM IN ASC

The problem of classification in ASC has been seen until now as that of assigning to an acoustic scene a label which corresponds to one of a closed set of classes. If the classifier knows only two outputs in the training set (for example *car* and *office*), it will classify any other scenes as one of the two, even when the scene does not correspond to either a *car* or an *office* (e. g. a train). From an application point of view, this approach will make meaningless assignments or decisions: examples are applications that automatically switch the ring-tone to silent mode in the *office* might trigger as well in the *park* or in other environments which are not present in the original closed set of classes.

Common to all of the past work, is the evaluation of ASC systems in a closed-set scenario for which training data is available for each and every acoustic class which may be encountered during testing. This evaluation strategy does not reflect practical applications in which out-of-set data may be readily encountered. Without any facility to reject out-of-class acoustic data, its assignment to a target class will result in degraded classification performance. As such, the current closed-set approaches to the evaluation of ASC systems do not reflect the level of performance which could be expected in most practical applications. Surprisingly, no previous work has investigated ASC in an open-set scenario.

In the machine learning literature, this problem is referred to as *open set* classification [138], where incomplete information of the classes is presented at training time, and completely unseen classes can be encountered during testing. The concept of *open-set* can be taken into consideration by heterogeneous classifiers such as one class classifiers [139], SVMs [140] or CNNs [141]. In other words, *open set problems* are used to recognise a finite set of known classes which are a subset of a greater number of unknown classes. Recent works [141] in visual recognition problem show how easy it is to deceive a classifier with *unknown* images. In multi-classification problems, *unknown* images are classified in one of the classes learned during training. If not handled properly, this produces many false predictions in the sense that a (true) unknown class is predicted as an existing (false) one. A similar concept is valid for a ASC task where the number of classes between training and testing can vary significantly.

The possibility to reject *out-of-class* samples is particularly pertinent for applied research, in order to avoid false detections. The use of ASC in *real, practical* applications would not be possible without an open set model since the high rate of false positives may affect the final precision on the class and performance estimates would not reflect those obtained in practise. Moreover, modelling exclusively the target classes (instead of modelling the entire set of classes) has other relevant advantages: first, the required amount of non-target samples is not crucial, with a significant reduction of costs and effort to collect these data which is in any case practically infeasible; second, the computational cost (in terms of memory for storing the model and computation for deciding which is the most likely scene) is proportional only to the number of target classes.

Content of this chapter illustrates the limitations of closed-set evaluation, proposes a new classifier, protocol and metric after having reinterpreted the problem of open-set

evaluation into a detection problems. The remainder of the chapter is organized as follows: Sec. 9.1 defines the open-set problem; Sec. 9.2 presents a classifier tailored to an *open-set* classification; Sec. 9.4 reports experimental results, whereas Sec. 9.5 presents our conclusions and some directions for further work.

9.1 CLOSED VS. OPEN-SET

ASC systems are usually developed using large collections of heterogeneous data. The data are aligned to a taxonomy in order to organize the collection into a number of groups or sub-groups which together span the data domain [142]. The groups are referred to as 'classes' or 'contexts' which gather together subsets of data which share similar characteristics. Examples are the classes *car*, *office* and *park*, all of which exhibit their own distinguishable characteristics.

The ASC task then involves the development of a statistical pattern recognition system whose aim is to predict the class to which an unlabeled sample should be assigned. A general approach to ASC thus involves the comparison of data samples to models of each acoustic class. When the universe of classes is exclusively predefined, and thus each sample must necessarily be assigned to one of the classes within, then the task is referred to as being *closed-set*. All existing ASC datasets and evaluations follow such a closed-set paradigm [6].

It is argued here that most practical applications are indeed uncontrolled and thus ASC solutions must necessarily be able to handle out-of-class data. Such an open-set system is easily realized with the addition of a *garbage* class to which should be assigned all acoustic data deemed insufficiently close to any of the other defined classes. Evaluation can then include out-of-class data. Examples for the previously described application could include *street*, *train* or *supermarket* noise. Out-of-class data should be as broad as necessary in order to reflect the practical application. The union of pre-defined and out-of-class data then makes up the entire acoustic universe.

9.1.1 *The concept of openness*

While the concept of closed and open-set problems is now clearly defined, the need to evaluate ASC performance in an open-set scenario leads to a *relative* concept of openness. An ASC system is designed to classify a number of *target* classes. In addition to the target classes there is a number of *known* negative classes. Any data sample not in either of these two classes is designated as a member of the *unknown* class. This arrangement is illustrated in the Venn diagram of Fig. 51. Formally, an open-set evaluation will thus involve some combination of t target classes, k known negative classes and u unknown negative classes. Their values are set according to an evaluation scenario or protocol as follows: a *training* dataset is composed of data from classes t and k while a *testing* dataset combines data from known classes t and k with additional data from unknown classes u.

The need for evaluation and the particular scenario impose some constraints on the values of t, k and u. While u is, by its very definition, unbounded, the evaluation of ASC systems can necessitate the definition of a notionally finite number of unknown classes; the value of t, k and u can reflect the difficulty of an evaluation. Tasks involving greater values of u and k relative to t are comparatively more difficult than tasks with smaller values. In particular, unknown negative classes are comparatively more difficult to handle than known negative classes. Related work [138] defines a measure, referred to as 'openness', which reflects the

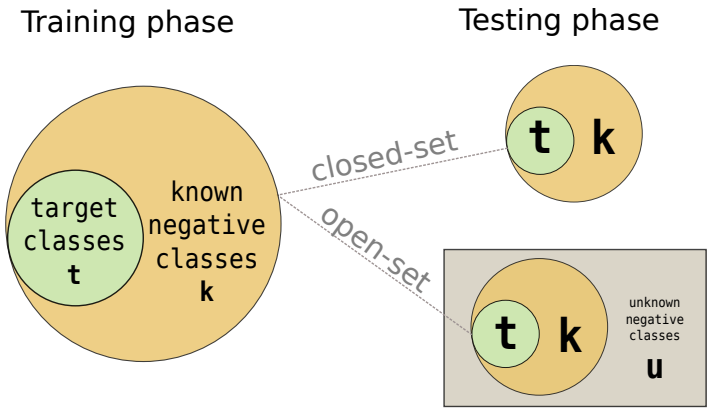


Figure 51: The universe of acoustic classes. Representative data from target and known negative classes $t \cup k$ are used for training. Representative data from unknown classes is used only in an open-set evaluation. One possible combination of acoustic classes may be t = car, k = office, park, u = street, train, supermarket.

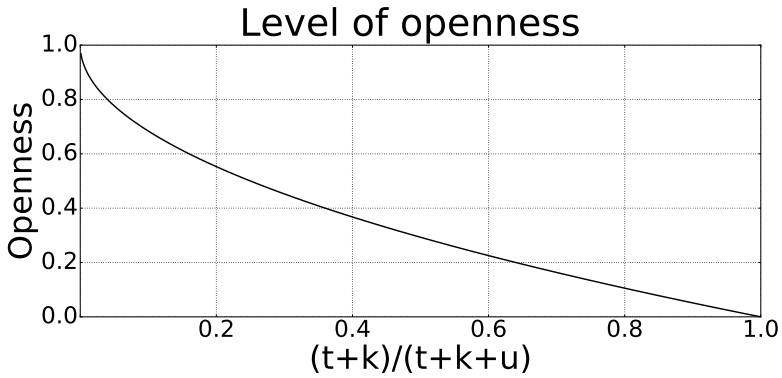


Figure 52: A plot of openness against the ratio of the number of training classes ($t+k$) and testing classes ($t+k+u$) according to Eq. 34. The openness increases as the number of unknown negative classes u increases.

difficulty of such a classification task. Drawing upon the aforementioned original work, a measure of openness is here expressed in terms of t, k and u as:

$$\text{openness} = 1 - \sqrt{\frac{t+k}{t+k+u}}. \tag{34}$$

An openness of 0 infers a closed-set problem, while an openness of 1 an entirely open problem. The square root tempers rapid increases in openness with only moderate u.

The relationship between the openness and the number of training classes $t+k$ and testing classes $t+k+u$ is illustrated in Fig. 52. Given a fixed number of targets t, the level of openness depends on k and u: when $u \gg k$, the level of openness will tend to 1; when $u \approx 0$ the level of openness will tend to 0. According to this assumption, the openness value relates to u, the number of unknown classes presented during testing.

While publicly available datasets for ASC do not preclude an open-set evaluation, standard evaluation protocols are all closed-set ($u = 0$). The second and third rows of Tab. 14 illustrate the openness of the standard, closed-set evaluation protocols for the DCASE 2013 and Rouen 2015 datasets [33]. Illustrated in the lowest five rows of Tab. 14 are different levels of openness for non-standard protocol adaptations which are discussed later.

Dataset	t	k	u	openness
DCASE 2013	10	0	0	0%
Rouen 2015	19	0	0	0%
DCASE closed-set	1	9	0	0%
Rouen closed-set	1	18	0	0%
DCASE open-set (4 targets)	4	4	2	10%
DCASE open-set (1 target)	1	4	5	29%
Rouen open-set (4 targets)	4	4	11	35%
Rouen open-set (1 target)	1	4	14	48%
Rouen open-set (1 target)	1	1	14	67%

Table 14: Examples of openness for two well-known datasets, standard closed-set ($u = 0$) and non-standard open-set ($u > 0$) protocols. Openness then varies as a function of the number of target classes t, known negative classes k and unknown negative classes u.

9.2 A CLASSIFIER TAILORED TO OPEN-SET

This section introduces the application of a SVM-based one-class classifier in the context of ASC, better suited to open-set classification.

9.2.1 *Support vector data description*

So-called one class SVM approaches have been investigated in the context of many different open-set problems, including image anomaly detection [143], machine fault detection [144] and spoofing detection for speaker verification [92]. One particular approach, referred to as support vector data description (SVDD), learns a hypersphere in which target samples are contained [145]. The goal is to represent target data within the smallest possible hypersphere volume. By using target data only for training purposes, SVDD avoids overfitting to known negatives and thus offers greater generalization to unknown negatives in an open-set scenario.

The hypersphere is characterised by its centre $\boldsymbol{a}$ and radius R which are adjusted to contain a percentage of training data $\mathcal{X}$. Based upon the intuition that possible errors will be reduced by minimizing the volume within the hypersphere, parameters $\boldsymbol{a}$, R and $\boldsymbol{\xi}$ are learned to minimize the following function with constraints:

$$\begin{aligned} &\min_{R,a,\xi} R^2 + C\sum_{i}^{N} \xi_i \\ &\text{s.t.} \quad \|\boldsymbol{x}_i - \boldsymbol{a}\|^2 \leqslant R^2 + \xi_i, \quad \xi_i \geqslant 0, \quad \forall i \end{aligned} \tag{35}$$

where $\boldsymbol{x}_i$ is the i^{th} sample of N training target samples, ξ_i is a penalty factor associated to each $\boldsymbol{x}_i$ and C is the importance associated to these penalty factors. The ξ_i variable adds a distance to data sample with a controlling factor C. C reflects the trade-off between the hypersphere volume and the percentage of training data contained within it. When $C < 1$, samples with a corresponding ξ_i will be allowed to remain outside the hypersphere without affecting the optimisation.

The minimisation problem with constraints in Eq. 35 is transformed into an unconstrained problem by using the Lagrange method [145]:

$$L(R, \boldsymbol{a}, \xi, \boldsymbol{\alpha}, \boldsymbol{\gamma}) = R^2 + C\sum_i \xi_i - \sum_i \alpha_i\{R^2 + \xi_i - (\boldsymbol{x}_i^2 - 2\boldsymbol{a}\boldsymbol{x}_i + \boldsymbol{a}^2)\} - \sum_i \xi_i\gamma_i \tag{36}$$

with the Lagrange multipliers $\alpha_i \geqslant 0$ and $\gamma_i \geqslant 0$. The maximum of the Lagrange function L is found by setting partial derivatives of R, $\boldsymbol{a}$ and ξ in Eq. 36 to 0, leading to the following constraints:

$$\begin{cases} \dfrac{\delta L}{\delta R} = 0: & \displaystyle\sum_i \alpha_i = 1 \\ \dfrac{\delta L}{\delta \boldsymbol{a}} = 0: & \displaystyle\boldsymbol{a} = \sum_i \alpha_i \boldsymbol{x}_i \\ \dfrac{\delta L}{\delta \xi_i} = 0: & \gamma_i = C - \alpha_i, \quad \forall i \end{cases} \tag{37}$$

The last constraint can be rewritten as $\alpha_i = C - \gamma_i$. Instead of putting a new constraint, α_i is used to obtain $\gamma_i = C - \alpha_i$ so that $\boldsymbol{\gamma} \geqslant 0$ is satisfied. The new constraint is simplified as $0 \leqslant \alpha_i \leqslant C$. Finally, by substituting constraints of Eq. 37 in Eq. 36, L assumes a quadratic form:

$$\begin{aligned} L(R, \boldsymbol{a}, \xi, \boldsymbol{\alpha}, \boldsymbol{\gamma}) = \; & R^2 + C\sum_i \xi_i - \sum_i \alpha_i R^2 - \sum_i \alpha_i \xi_i - \sum_i \gamma_i \xi_i \\ + \; & \sum_i \alpha_i \boldsymbol{x}_i^2 - 2\sum_i \alpha_i \boldsymbol{a}\boldsymbol{x}_i + \sum_i \alpha_i \boldsymbol{a}^2 \\ = \; & \cancel{R^2} + C\sum_i \cancel{\xi_i} - \cancel{R^2} - \sum_i \cancel{\alpha_i \xi_i} - C\sum_i \cancel{\xi_i} + \sum_i \cancel{\alpha_i \xi_i} \\ + \; & \sum_i \alpha_i \boldsymbol{x}_i^2 - \cancel{2}\sum_{i,j} \alpha_i \alpha_j \boldsymbol{x}_i \boldsymbol{x}_j + \cancel{1}\sum_{i,j} \cancel{\alpha_i \alpha_i \boldsymbol{x}_i \boldsymbol{x}_j} \\ = \; & \sum_i \alpha_i \boldsymbol{x}_i^2 - \sum_{i,j} \alpha_i \alpha_j \boldsymbol{x}_i \boldsymbol{x}_j \end{aligned} \tag{38}$$

Interestingly, the dual form of Eq.35 becomes a maximisation problem with respect to the Lagrangian $\boldsymbol{\alpha}$

$$\begin{aligned} \max_{\alpha} \quad & \sum_i \alpha_i \boldsymbol{x}_i^2 - \sum_{i,j} \alpha_i \alpha_j \boldsymbol{x}_i \boldsymbol{x}_j \\ \text{s.t.} \quad & 0 \leqslant \alpha_i \leqslant C. \end{aligned} \tag{39}$$

The solution to Eq. 39 gives the set of α_i which characterizes the SVDD model (the centre of the hypersphere). The Lagrangian α_i satisfies one of the following conditions:

- for $\alpha_i = 0$, data sample $\boldsymbol{x}_i$ will be within the hypersphere;
- for $0 < \alpha_i \leqslant C$, $\boldsymbol{x}_i$ will be on the boundary or outside the boundary. Data samples lying on or beyond the boundary are referred to as SVs;
- for $0 < \alpha_i < C$, $\boldsymbol{x}_i$ will identify support vectors which lie on the boundary. They are referred to as boundary support vectors (BSVs).

Thus, the radius of the hypersphere is the distance from its centre to one of the BSVs, $\boldsymbol{x}_k$:

$$R^2 = \|\boldsymbol{x}_k - \boldsymbol{a}\|^2 = (\boldsymbol{x}_k \cdot \boldsymbol{x}_k) - 2\sum_i^N \alpha_i(\boldsymbol{x}_i \cdot \boldsymbol{x}_k) + \sum_{i,j}^N \alpha_i\alpha_j(\boldsymbol{x}_i \cdot \boldsymbol{x}_j). \tag{40}$$

A data sample lies within the hypersphere if its distance from the centre is less than the radius. A test sample $\boldsymbol{z}$ is within the hypersphere (so accepted as *target* sample) when:

$$\begin{aligned} &R^2 - \|\boldsymbol{z} - a\|^2 > 0 \\ &R^2 - (\boldsymbol{z} \cdot \boldsymbol{z}) + 2\sum_i^N \alpha_i(\boldsymbol{z} \cdot \boldsymbol{x}_i) - \sum_{i,j}^N \alpha_i\alpha_j(\boldsymbol{x}_i \cdot \boldsymbol{x}_j) > 0. \end{aligned} \tag{41}$$

9.2.2 *Gaussian kernel*

Data samples are mapped into a higher dimensional space where the boundary is optimal in describing the target class. As for regular SVMs, the kernel trick avoids the need to compute explicit coordinates in the higher dimensional space [146]. With the dual form, the centre is not calculated explicitly, since it can be replaced by the inner products between all pairs of data samples $\boldsymbol{a} = \sum_i^N \alpha_i \boldsymbol{x}_i$ (from constraints in Eq. 37). The most flexible kernel function in many real-case scenarios, and that used here, is the Gaussian kernel [147, 148] expressed as K:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\sigma^2}), \tag{42}$$

with σ which indicates the variance of the Gaussian distribution. The matrix $\boldsymbol{K}$ indicates all-pair distances between training samples. The term $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ does not depend upon the position of the data from the sphere centre. When two samples are closely located such that $\boldsymbol{x}_i \simeq \boldsymbol{x}_j$, then $K(\boldsymbol{x}_i, \boldsymbol{x}_j) \to 1$; when two samples are well separated such that $\boldsymbol{x}_i \neq \boldsymbol{x}_j$, then $K(\boldsymbol{x}_i, \boldsymbol{x}_j) \to 0$. The distance depends on the choice of σ . Tab. 15 presents the link between σ, K and the Lagrangian α, while the resulting decision boundaries are sketched in Fig. 53.

Fig. 53 shows the importance of σ in the generalisation capability of the classifier: σ plays the role of normaliser (amplifying or attenuating) the distance between any $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. If the distance between them is larger than σ, the kernel value tends to 0. If σ is small, instead, values of relatively fewer samples will influence the distance. In other words, smaller σ tends to make a locally optimised classifier, while larger values of σ tend to build a more generalised classifier. In the case of a Gaussian kernel, $K(\boldsymbol{x}_i, \boldsymbol{x}_i) = 1$. By setting the first term in Eq. 39 to 1 ($\sum_i \alpha_i = 1$) and by splitting the second term into two components where $i = j$ and $i \neq j$, the following expression is obtained:

$$\max_{\alpha} \quad 1 - \sum_i \alpha_i^2 - \sum_{i \neq j} \alpha_i\alpha_j\boldsymbol{x}_i\boldsymbol{x}_j. \tag{43}$$

The relationship between σ, kernel matrix $\boldsymbol{K}$ and Lagrangian $\boldsymbol{\alpha}$ is clarified as follows:

1. in an overfitting case, a small σ produces $K(\boldsymbol{x}_i, \boldsymbol{x}_j) \to 0, i \neq j$. The impact of term $\sum_i \alpha_i^2$ over the maximisation is minimal when many α_i assume small values. This also means that a large portion of the training samples will become SVs. A larger number of SVs corresponds to a more complex boundary and a higher risk of overfitting (Fig. 53 on the left);

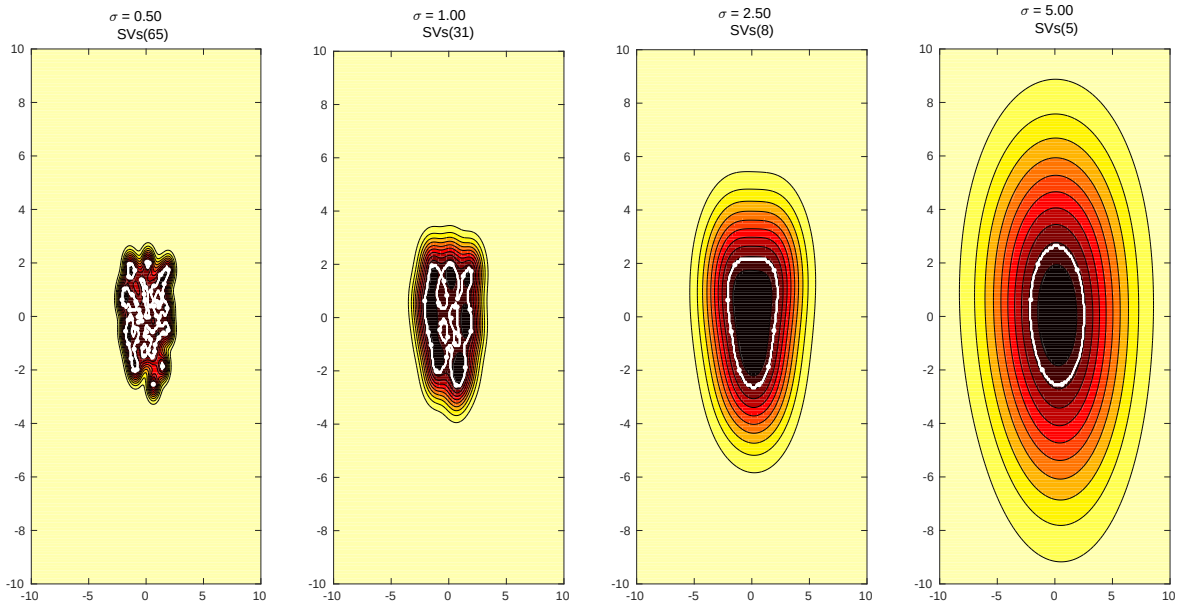


Figure 53: An illustration of how σ influences the number of SVs and the shape of the sphere. The data has been artificially created using a Gaussian distribution of 100 samples of 2 dimensions. SVs are indicated with white dots; the decision boundary is the line in white. Contour describes the decision function values which vary from a target (in black) to non-target class (in white), with all intermediate values.

Case	**Overfitting**	**Underfitting**
Variance	$\sigma \to 0$	$\sigma \to \infty$
Kernel distance	$K(\boldsymbol{x_i}, \boldsymbol{x_j}) \to 0$	$K(\boldsymbol{x_i}, \boldsymbol{x_j}) \to 1$
Lagrangian	$\alpha_i = \frac{1}{N}$	Few $\alpha_i \neq 0$

Table 15: The following scheme represents the influence of σ on the kernel distances and therefore on Lagrangian $\boldsymbol{\alpha}$.

2. in an underfitting case, a large σ produces $K(\boldsymbol{x}_i, \boldsymbol{x}_j) \to 1$. The function is maximised when a significant fraction of $\alpha_i = 0$. This means fewer SVs and a simpler boundary (Fig. 53 on the right).

9.3 GRID-SEARCH STRATEGIES

The generalisation capacity of the SVDD algorithm depends on the choice of model parameters (C, σ). Standard approaches compute an error estimation on the test set which reflects the level of performance expected during evaluation. Nevertheless, this estimation is not reliable when conditions vary from validation to evaluation set. Open-set evaluation, by definition, expresses this difference in the composition of non-target classes. Therefore the error estimation (and related parameters tuning) plays an important role in any open-set scenario and it is further discussed herein. This error is defined as ϵ_I, or error of the first kind or false negative rate. On the other hand, without knowing any information about the

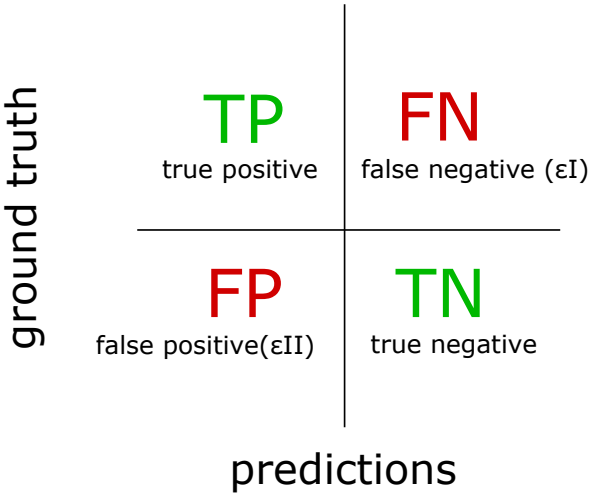


Figure 54: The confusion matrix while classifying an object in one-class classification problem. The fraction of the target samples classified as outliers is the false positive rate (ϵ_I), whereas outliers labelled as targets the false positive rate (ϵ_{II}).

non-target samples composition, the estimation of the false positive rate, ϵ_{II}, is not reliable. The two types of error are illustrated in Fig. 54.

In binary SVM classifiers, samples from both target and non-target classes are represented and used to build the decision boundary between the 2 sides of the feature space. In open-set classification, instead, only target samples are available while the non-target samples are not representative of the non-target distribution. It is much harder to select the boundary in this sense and how tightly fitted it should be. The estimation of the false positive rate (ϵ_{II}) is problematic: in fact, the SVDD estimates solely the number of target samples which are rejected (the false negatives). For SVM-based classifiers, the ratio between the number of SVs and the training set size (referred to as *SV ratio*) produces a reliable estimation of false negatives [70] (Sec. 3.2). The SV ratio is used also for the SVDD classifier, which is based upon the SVM theory.

As an example of the application of SV ratio to SVDD, the curves of the false negative rate (ϵ_I) and the SV ratio are illustrated in Fig. 55. The samples are trained with using the SVDD classifier on a 2-D Gaussian distribution of 50 samples and tested on 200 samples from a uniform distribution. The condition $C = 1$ indicates that all the training samples are considered *target* samples. As demonstrated in Fig. 53, σ influences the number of SVs and therefore the complexity of the classifier: the number of SVs tends to decrease as σ increases. This behaviour is confirmed by experiments in Fig. 53 where, at higher values of σ, fewer SVs are needed to model the SVDD hypersphere boundary.

Hence, C influences the number of target samples rejected and σ the number of SVs. These considerations allow to specify an expected rate ϵ_I for C, σ in the case of a Gaussian kernel. While the SV ratio expresses a measure of the false negative rate (ϵ_I), the false positive rate (ϵ_{II}) cannot be reliably estimated for the SVDD. In fact, non-target samples are poorly represented in an open-set scenario, as pointed out by [145], yet they are crucial to the SVDD generalisation capability. Thus, the criterion to select the optimal (C, σ) pair depends upon the availability of non-target samples. Two different criteria are then proposed to estimate ϵ_{II}: one which does not depends on the presence of non-target samples; the other which does employ the non-target samples when available. The two criteria are described in the following.

MIN #SV MAX RADIUS The problem of autonomously tuning C and σ without any knowledge of the non-target class is investigated by [149]. The optimal (C, σ) pair is found

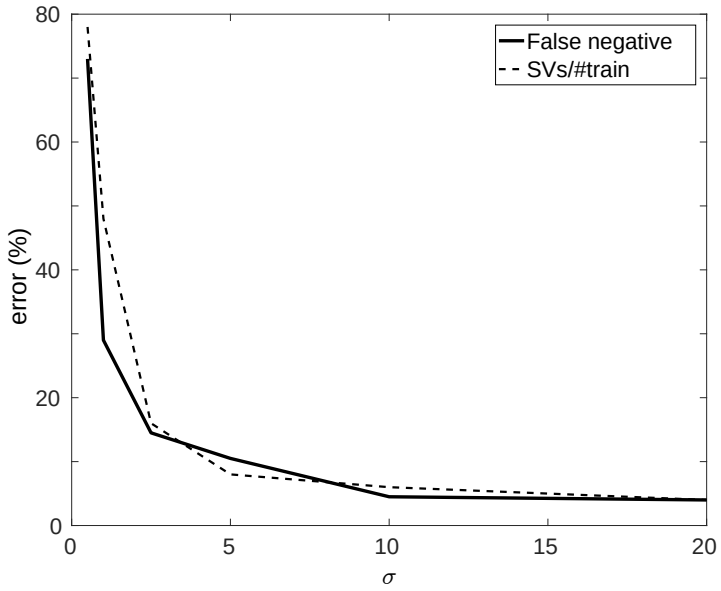


Figure 55: Estimation of the false negatives (target samples which have been rejected). The ratio of SVs on the boundary and the number of training samples is reported in dashed black line while the false negative rate (computed over the test set) is in solid black line.

based on an optimization criterion. This criterion is referred to as λ_{radius}: on one hand, the radius R should tend to 1 so as to produce a precise hypersphere boundary (thereby producing a lower false positive rate); on the other hand, a SV ratio which tends to 0 indicates a lower false negative rate. The λ_{radius} criterion is expressed as:

$$\lambda_{radius} = \sqrt{\left(\frac{\#SV}{N}\right)^2 + (1-R)^2}. \tag{44}$$

Both $\frac{\#SV}{N}$ and R values are in the $[0, 1]$ range. The radius term is expressed by $1 - R$ so that the minimisation of λ_{radius} expresses the minimisation of $\frac{\#SV}{N}$ and maximisation of R.

The λ_{radius} criterion has some advantages: the tuning does not require any information of the non-target distribution and does not depend on the number or type of classes in the training-validation set. However, in situations where the separability is critical, it can lead to sub-optimal solutions.

MIN #SV MAX AUC Some different approaches to grid-search may exploit the presence of non-target samples in the training set to automatically select the best pair of parameters. The λ_{AUC} criterion includes a notion of validation error, computed on a sub-set of the training set (the validation) with the AUC, the area under the receiver operating characteristic (ROC) curve:

$$\lambda_{AUC} = \sqrt{\left(\frac{\#SVs}{N}\right)^2 + (1-AUC)^2}, \tag{45}$$

where #SVs corresponds to the number of SVs and N is the cardinality of the target class. Obviously, the quality of this estimation depends on the number and representativeness of non-target samples. Therefore, λ_{AUC} defines a trade-off between

an estimation of the classifier complexity ($\frac{\#SVs}{N}$ term) and the SVDD performance expressed with the AUC metric.

The evaluation of λ_{radius} and λ_{AUC} performance is reported in Sec. 9.4.4.

9.4 FROM CLASSIFICATION TO DETECTION: EXPERIMENTAL RESULTS

This section reports an evaluation of ASC in open and closed-set scenarios. The evaluation is performed in a single-class detection mode. Detection, as opposed to classification, allows for assessment with a comparatively simple metric [150] and also gives a more reliable indication of performance which is less influenced by the number of classes in the dataset. It is stressed, however, that this approach does not preclude multi-class classification which could be implemented straightforwardly with multiple detectors [139]. Furthermore, the choice of a detector has been driven by the applicability in unconstrained scenarios, where only a single target class is of interest and the other classes have to be rejected.

9.4.1 *Implementation details*

MFCC features are extracted from frames of 32ms with a 16ms overlap and accumulated to form audio segments of 4s segments overlapped by 2s. The frequency range is set to $[0, 8000\text{Hz}]$. Mean and standard deviation statistics are extracted over each segment (without the C0) thereby creating a 25-dimensional feature vector. MFCCs use *rastamat* library [151] with default settings. SVM and SVDD classifiers are both implemented using the *libSVM* library [152] using a Gaussian kernel. The parameters for this kernel are tuned independently for each classifier. Finally, feature vectors are normalized according to the *z-score* method [153].

9.4.2 *Datasets and protocols*

The DCASE 2013 and Rouen 2015 datasets are used for evaluation. For the DCASE 2013 database, it has been used the development set for training/validation and the evaluation set for testing. For Rouen 2015, testing is performed using a *5-fold* cross-validation. In both cases, evaluation involves a gradual transition from closed-set to progressively more open-set configurations. Reported first are results for a closed-set evaluation which corresponds to the configurations of the second and third rows of Tab. 14.

Acoustic class models are learned independently for each target. SVM training is performed using data from both $t = 1$ target class and k known negative classes. In contrast, the SVDD classifier is trained using target class data alone. In the case of the λ_{AUC} criterion, SVDD exploits the presence of non-target samples to estimate the AUC metric and to select the best parameters. In order to vary the degree of openness, the number of known negative classes k is varied in both cases from 1 to $C - 1$, where C is the total number of classes involved in the evaluation ($C = 10$ for DCASE 2013 and $C = 19$ for Rouen 2015).

Testing is performed using varying quantities of data from the whole acoustic universe encompassing t, k and u. When $k = C - 1$, the evaluation is closed-set. The number of unknown acoustic classes in this case is $u = C - t - k = 0$. To better illustrate the closed-set protocol, consider the detection of the *bus* class using the Rouen dataset where $C = 19$. If the number of known negative classes is set to $k = 4$, then the number of unknown negative classes is $u = 14$. According to Eq. 34, this setup corresponds to an openness of 48% as illustrated in the penultimate row of Tab. 14.

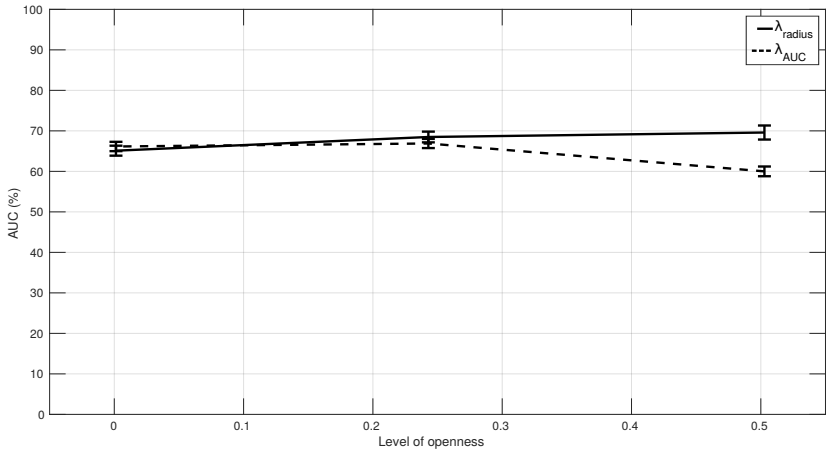


Figure 56: λ_{radius} and λ_{AUC} comparison, respectively in solid black line and dashed black line. AUC metric refers to the average over all classes at the same level of openness. Results relate to DCASE 2013 dev set.

In practice, the performance of the SVM classifier which exploits known negative data will depend on exactly what composes the k known negative classes. Accordingly, for each level of openness 10 random selections of k known negative classes is performed. AUC mean and standard deviation are reported.

9.4.3 *Detection metric*

Classification accuracy is the standard metric for the evaluation of ASC systems. However, the intrinsic limitations of classification accuracy [32] mean it is ill-suited to open-set problems. Consequently, the area under the curve (AUC) metric is preferred instead. The AUC is not influenced by the ratio of target and negative classes and is threshold independent [154].

The latter graphically expresses the binary separability with respect to a threshold. The thresholded values are referred to as *scores*. Scores describe the separability in terms of continuous values (e. g. probabilities, distances, similarity measures). For the SVM classifier, scores are extracted using the Platt method [155] which transforms the distances from the hyperplane into class probabilities; for the SVDD classifier, the scores are calculated according to the distance to the hypersphere radius $R^2 - ||z - a||^2$. The AUC is then computed for each classifier and for different levels of openness, averaged over all $t = 1...C$ classes and 10 random compositions of k known negative classes.

9.4.4 *Grid-search results*

For the SVM classifier, parameter tuning is performed using cross-validation based on the highest validation accuracy. For the SVDD classifier, parameters are optimised by minimizing one of the two proposed criteria λ_{radius} or λ_{AUC}.

Results for the best criteria for tuning (C, σ) are reported in Fig. 56. Implementation details, protocols and detection metrics follow those presented in Sec. 9.4.1 and 9.4.2. λ_{radius} and λ_{AUC} are computed over the validation set, a randomly 30% selection of the training set. To avoid bias between segments coming from the same recording, the split is always performed at recording level. Fig. 56 reports the AUC metric averaged over all classes at different levels of openness.

Results for the DCASE 2013 development dataset confirm the intuition that λ_{radius} does not depend on the training set composition whereas λ_{AUC} is influenced by the level of openness. When a good representation of the non-target classes is available (i. e. therefore

at lower levels of openness), the λ_{AUC} provides higher performance in terms of AUC. In the experimental works reported in the next section, the λ_{radius} criterion is preferred because of its non-dependency on the non-target data composition. However, for specific applications, the λ_{AUC} may provide better performance.

9.4.5 *SVM vs SVDD*

Results are illustrated for the DCASE 2013 and the Rouen 2015 datasets in Figs. 57 (a) and (b) respectively. Results for the SVM classifier are illustrated by dashed-blue profiles. Those for the SVDD classifier are illustrated by solid-red profiles. For each level of openness, AUC results are averaged over all classes with the same level of openness. Vertical bars in Fig. 57 reflect the AUC standard deviation over these classes.

Similar trends are observed for both datasets. As the openness increases, the performance of the SVM classifier deteriorates, falling from 95% to 60% for the DCASE 2013 dataset and from 90% to 50% for the Rouen 2015 dataset. In contrast, results for the SVDD classifier remain relatively stable for both datasets, measuring in the order of 80% and 85% of AUC for the DCASE 2013 and Rouen 2015 datasets respectively.

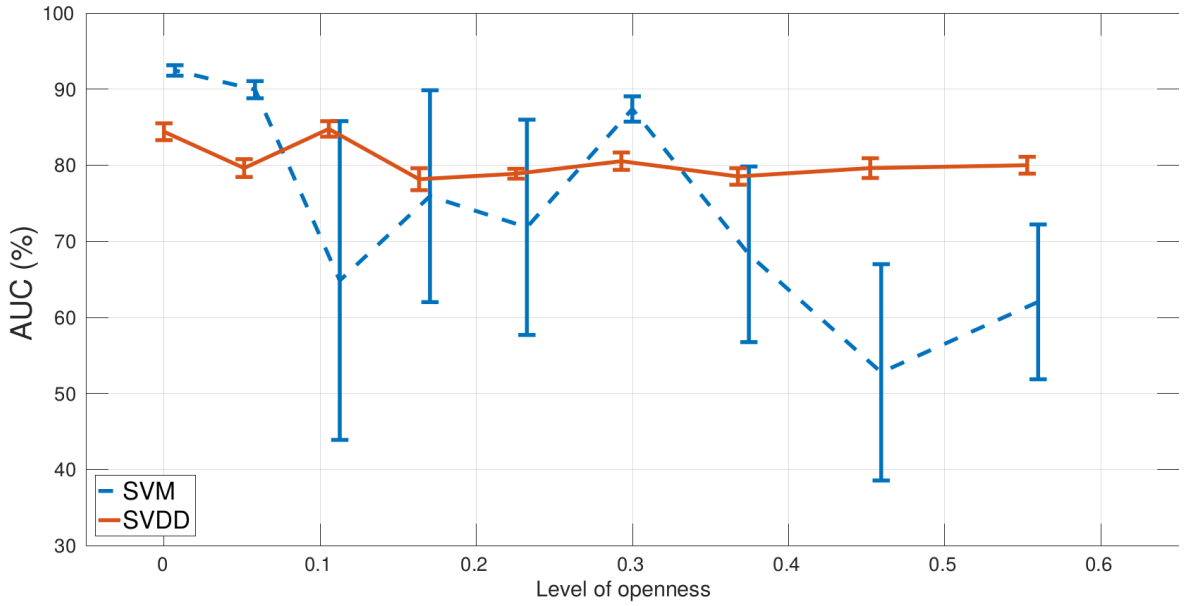
Results in Fig. 58 illustrate separately the AUC for each class in the Rouen 2015 dataset for an openness of 0.67. Consistent with results illustrated in Fig. 57, the SVDD classifier outperforms the SVM classifier. Of greater interest here, however, is the variation in performance for different compositions of k known negative classes, again illustrated in terms of standard deviation with vertical bars. While the performance of the SVM classifier is impacted by a specific combination of k known negative classes, that of the SVDD classifier is relatively unaffected.

Fig. 59 reports, instead, the results of two systems characterised by different feature extraction methods but having the same SVDD classifier. Features are: i) MFCC features detailed in 9.4; local binary patterns (LBPs), root mean square (RMS) and band energy ratio (BER) (`LBP+RMS-based+BER`). For further details about features `LBP+RMS-based+BER`, refer to Sec. 5.3. The system based on `LBP+RMS-based+BER` features outperforms that based on MFCC features in the case of a small number of classes, such as *bus*, *metro-paris*, *quietstreet* and *tubestation*. For others (e. g. *airplane*, *metro-rouen*, *pedestrian street* and *train*), the level of AUC for the two systems is similar. For the remainder of the classes, the SVDD-MFCC system outperforms the other system.

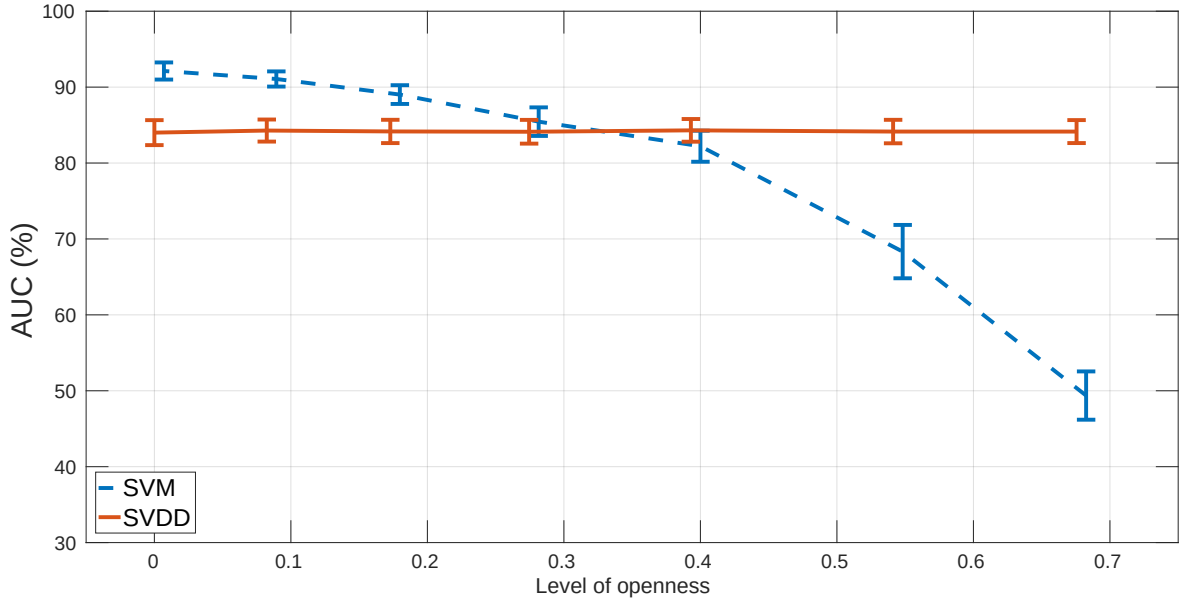
9.5 CONCLUSIONS AND FUTURE DIRECTIONS

This chapter reports the first attempt to develop an approach to acoustic scene classification (ASC) for a practical, open-set scenario. A traditional ASC classifier is shown to outperform an open-set classifier in a largely closed scenario. When the level of openness increases, however, performance degrades rapidly, whereas the performance of the newly proposed approach to open-set ASC remains stable. The SVDD classifier learns a hypersphere from target data only. While using target data only for training, this classifier is less susceptible to overfitting to known negative data and is thus more reliable in the face of unknown negative data. A new approach based on a detection formulation, a new protocol and metric are also introduced.

A further contribution relates to the importance of model parameter tuning. Two methods are compared: one based on a target-based criterion and a second, aware of non-target samples. Depending on the type of ASC applications, one criterion may be preferred to the other : under a 0.1 level of openness, making use of the entire training set (target and



(a)



(b)

Figure 57: Plots of the area under the receiving characteristic curve (AUC) against openness for (a) DCASE 2013 evaluation set and (b) Rouen 2015 datasets for SVM (dashed-blue profiles) and SVDD (solid-red profiles) classifiers. Standard deviation is illustrated with vertical bars.

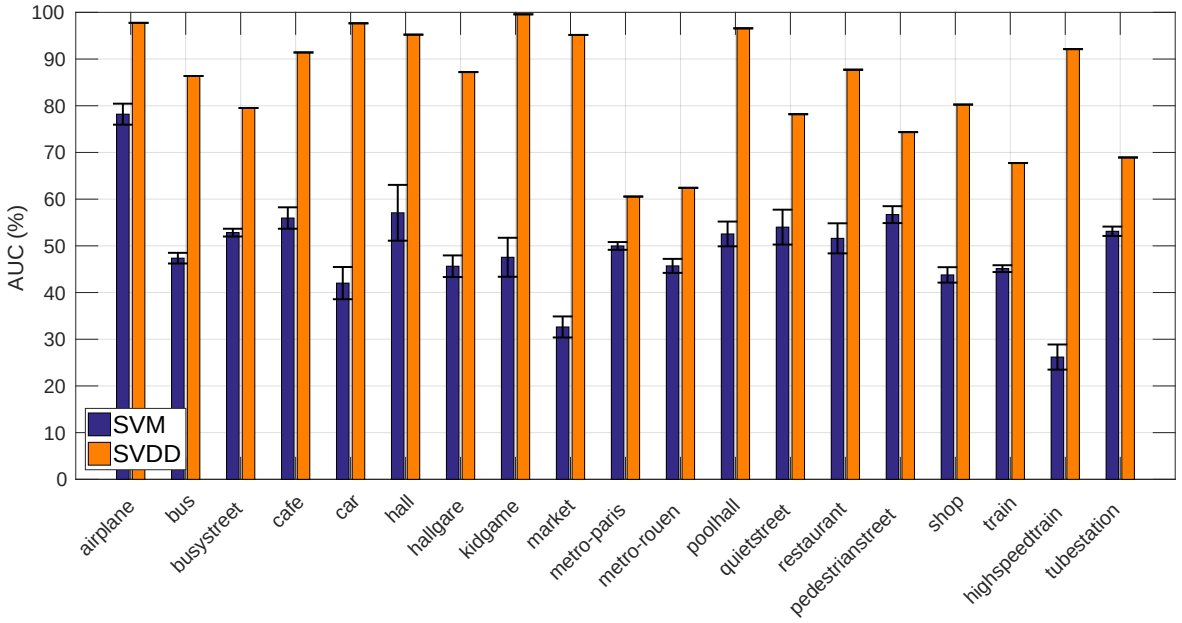


Figure 58: Individual class AUC results for the SVM and SVDD classifiers for the Rouen 2015 dataset with an openness of 0.67.

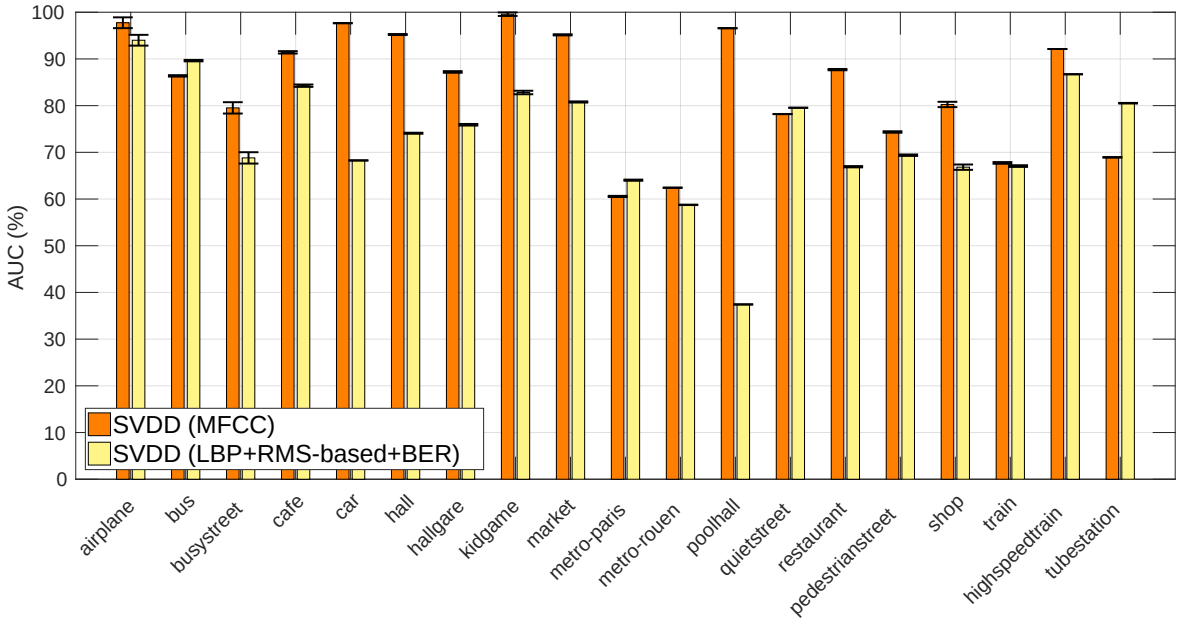


Figure 59: Individual class AUC results for SVDD classifier and two different feature extraction methods: MFCC (dark-red bar profiles) and `LBP+RMS-based+BER` (light-yellow bar profiles). The referred dataset is Rouen 2015 with an openness of 0.67.

non-target, if available) is beneficial; when the uncertainty is high, a criterion based solely on training data seems more robust.

The performance of the SVDD algorithm is correlated to the type of features used to describe each acoustic scene. As an example, LBP-based features show better performance for some classes whereas MFCC-based features lead to better reliability in the case of some other classes. More generally, open-set approaches can be designed with any classifier, including deep learning approaches. Given recent work which shows the vulnerability of deep learning architectures [156] to specifically designed samples, an open-set evaluation is needed. Domains as image classification [157] and face verification [158] have started questioning closed-set evaluations. There is evidence that current deep learning approaches show too optimistic performance and they are not robust to unknown samples during testing [141].

In ASC, inter and intra-class variability is so high that the open-set scenario has to be taken into account. Future ASC evaluation should consider this scenario, since it provides an evaluation framework which is closer to reality. The SVDD classifier is a possible solution to the open-set problem, but other open-set aspects should be investigated in future research:

- a better feature characterization of each acoustic scenes (e. g. using CNN architectures to automatically extract features from data);
- the integration of the open-set risk in the error minimization (e. g. replacing the *soft max* function with an *open max* function, tailored to open-set [141]);
- the exploitation of non-target samples, when available (e. g. SVDD which takes advantages of non-target samples [159]);
- novelty detection of unknown samples with the automatic definition of new classes (e. g. novelty detection based on SVDD distances).

Given that the predominant ASC use-case scenario is open-set in nature, it is hoped that the proposed perspective on ASC will be adopted by the research community in the future.

10

CONCLUSIONS AND FUTURE WORK

The scope of this thesis has been to investigate acoustic scene classification (ASC), with the goal being to deliver context-awareness for low power devices carried by people during daily activities. *Context-awareness* has broad appeal, e. g. for controlling the frequency of notification alarm with respect to the context (e. g. in the *car* or at *home*); adapting phone call volume to the surrounding environment or enabling *context-based* configurations for hearing aids. The implementation of ASC on *embedded* devices is a strategic choice since it respects user privacy; there is no need to communicate potentially private, sensitive acoustic information (speech and audio events/scene) to external services. Furthermore, unreliable data connections and power implications add to the appeal of performing ASC *locally* on the device.

Given this emphasis on embedded applications, the research performed through this thesis study had to consider two aspects: on one side, the proposed ASC systems required a comparison to the current state-of-the-art performance; on the other side, the applicability to "real-world" products introduced practical constraints (e. g. always-listening and low-complexity systems). The motivation of this thesis, then, stems from reducing the gap between fundamental (Part 1) and applied research (Part 2).

With ASC being a recent field of study, standard methods borrow techniques from other related domains (such as MFCC features for speech or music genre recognition) without investigating ASC peculiarity. The analysis of two decades of ASC literature described in Chapter 2 shows that MFCCs are the most popular features. Nevertheless, the top performing systems [56, 33] used features specifically tailored to the ASC problem. These findings confirm that, while the ASC community can benefit from prior research in other domains, there is also a need for specific solutions.

Aside from fundamental research, a more *applicative* aspect should be considered when implementing ASC systems for real products (e. g. smartphones or hearing aids). These aspects include: *always-listening* systems which process and analyse sounds in a continuous streaming fashion; low-power systems which limit the computational power and the memory required to store models. Even if these product constraints may sound a limitation, they enlarge the current ASC research scopes, thereby introducing new areas of studies.

10.1 WHAT HAS BEEN DONE?

From the analysis of DCASE 2013 results (Chapter 2), standard MFCC-based features are found to be insufficiently discriminative to capture the complex spectro-temporal structure of an acoustic scene. The winning system of DCASE 2013 complemented information extracted with standard MFCCs with features capturing frame-level temporal recurrence. This new set of features captures the recurrence of MFCCs in the acoustic scenes. However, being still based on MFCCs, this method has several limitations such as a high dependence on scene energy, poor generalisation across multiple datasets and poor robustness to scene variations.

In order to have a first baseline, the winning system of DCASE 2013 (referred to as RNH) was re-implemented obtaining the same level of performance reported in [56]. Experimental results reported in Chapter 3 show that small changes in the feature parameters can impact drastically on performance. In particular, these changes can be grouped as follows: i) differences in the energy level; ii) differences in the frequency range; iii) differences in the segment length from which features are extracted.

Besides these considerations, optimised features seem be dependent on the composition of the scenes to be recognised. Chapter 4 reports visualisations and feature metrics which are designed to shed light on the relationship between features and acoustic scenes: t-SNE is a non-linear visualization technique which can be used to represent high-dimensional features in 2D mappings. This visualisation feedback helps in the design of new features. In addition, feature metrics such as the *Fisher* score and the Bhattacharyya distance complement the information coming from t-SNE visualisations by providing quantitative metrics which are independent of a particular classifier. Experimental work reported in Chapter 4 demonstrates that, depending on the type of scenes composing the database, certain features are better suited than others. As an example, the first MFFC coefficient (C0, which reflects the level of energy in the signal) is replaced by relative-measures of energy. These features are based upon variations in the root mean square values (RMS) and over the band energy ratio (BER). Results showed consistent performance over multiple databases.

A very relevant aspect discussed in Chapter 4, then, is the cross-database validation to test the generalisation of ASC systems. At the time when this thesis started, in 2014, only the DCASE 2013 database was publicly available. Moreover, the recording conditions of these datasets were far from the realistic ones recorded by mobile devices. The NXP dataset contains 30h of audio recordings collected from different scenes and forms a contribution of this thesis. The NXP dataset provides far much broader and various data; augments the diversity of recording conditions and better represents the heterogeneity of each acoustic scene.

Recent trends in the ASC literature [33] employ image-processing techniques on audio spectrogram. This suggests that spectro-temporal information is beneficial to ASC. Work presented in Chapter 5 demonstrates that local binary patterns (LBPs) could represent the entire scene with a unified descriptor. Results obtained with LBPs outperformed state-of-the-art methods on multiple datasets. The capture of spectro-temporal structure through spectrogram patterns represents a significant improvement with respect to traditional features for ASC.

Nevertheless, ASC is characterised by a significant inter and intra scene variability. With such variability, the design and tuning of LBP features is still correlated to the number and type of acoustic scenes. Every time a new class is introduced, features need to be re-optimised. Recent advances in deep learning techniques offer a promising alternative to the use of *hand-crafted* features as well as a suite of different approaches to automatic feature learning from complex input data (e. g. images and audio). The research presented in Chapter 6 describes the use of a convolutional neural network (CNN) based on a 2-channel input spectrograms (log-mel power spectrogram + first derivatives Δ) and specifically adapted to the ASC task. Like LBPs, this network topology captures local correlation in time-frequency domain. Results on publicly available DCASE 2016 dataset showed competitive results of this approach. The review of the systems submitted to the DCASE 2016 challenge in Chapter 7 reports a significant adoption of CNNs for ASC. This trend consolidates the hypothesis of using spectro-temporal techniques as suggested in Chapter 5 and 6.

Even so, current solutions in the literature did not investigate the impact of running ASC on *embedded* devices. This implementation imposes limitations in terms of memory storage, computational cost and real-time processing. A complete solution to solve the

aforementioned limitations is presented in Chapter 8. MFCC statistics (mean and standard deviation) are estimated in *real-time* with two recursive estimators. These two estimators work in *tandem* to better capture variation in the signal. To reduce the model size, a reduced complexity SVM model is obtained by clustering and decimating the training samples without a significant impact on performance.

Even if previous methods enable ASC systems to run on embedded devices, these systems poorly react to *unknown* classes during *in-field* evaluation. The rejection of unknown classes is recognised as a fundamental requirement in order to reduce false positives and then to make applications possibly *usable*. Common to all of the prior work is the evaluation of ASC systems in a closed-set scenario for which training data is available for each and every acoustic class which may be encountered during system use. It is argued in this thesis (Chapter 9) that ASC is an open-set problem by nature: realistic ASC applications should be able to recognise an acoustic scene among a set of unknown scenes (i. e. not seen during training phase). Possible solutions consist of only modelling the target class without creating a corresponding non-target. As an example of this idea, the support vector data description (SVDD) classifier tailored to open-set is proposed in Chapter 9, together with metrics (receiver operating characteristic (ROC) curve and corresponding area under the curve (AUC)) and protocols more suited to an open-set scenario. It is strongly believed that future ASC evaluations should consider this scenario, since it provides results which are close to those one could expect in "real-life".

10.2 WHAT CAN BE CONCLUDED?

The research on the *peculiarities* of ASC is the topic that the author has tried to investigate throughout the thesis. From the offered views to this subject, some general conclusions are derived. They are detailed as follows:

- ASC is a highly complex task from an acoustic and taxonomy point of view. For example, similar acoustic scenes could be classified under two different high-level concepts (e. g. *quiet street* and *park*) while the same concept may contain very different acoustic scenes (e. g. *car* contains sport car and electric car). A comprehensive dataset which captures such variation would be expensive and difficult to collect. In addition, obtaining an agreement from the community on a common taxonomy would also represent a big challenge;

- an acoustic scene has a weak temporal structure. Prominent sounds may appear in any order so that any methods which model a temporal evolution will not be suited to represent this temporally unstructured scene. Systems based on LBPs or CNN rely upon the presence of specific patterns rather than their temporal evolution. This idea can be seen under the *bottom-up* perspective (Chapter 2). A bottom-up perspective groups different methods under the common idea that low-level audio descriptors (in the case of LBPs and CNN, the audio patterns) compose the entire acoustic scene;

- an acoustic scene can be characterised by spectro-temporal patterns, which extract information from time-frequency representation. The nature of these patterns can be decided *a priori* (e. g. LBPs) or automatically extracted from the data (e. g. CNNs). What is different from traditional features (e. g. MFCCs), is the significant correlation in time and frequency showing that a unified descriptor in time and frequency can obtain a high level of performance;

- ASC is an open-set classification problem. Before performing any classification, a robust ASC system should first determine whether or not the scene is within the set

of known classes or label it as *unknown*. In this case, the ASC system would perform detection before classification.

10.3 On what should future research focus?

Research in the ASC domain still relies on a *supervised* scenario where labels and data must be provided. This supervised paradigm is very inefficient when the amount of training data represents very poorly the true variability of acoustic scenes. It is recommended that future research should investigate different approaches, which do not depend entirely on labelled data. In that sense, learning from weakly annotated data coming from other domains (video, audio event) is a possible option. From a similar perspective, continuous active learning approaches should be considered as an alternative to existing supervised ones. ASC solutions may have access (through a microphone) to an essentially infinite amount of unlabelled data, but labelling this data is expensive. Semi-supervised approach such as active learning [160] can select a subset of such data to label automatically. Once the sample has been labelled (from an interaction with the user or with other source of information), scene models can be re-trained or adjusted.

Another research track may involve *transfer learning* [161, 162] between related domains. For example a system designed to detect audio events could be used to classify scenes without complete re-training. In this case the goal is to train a complete ASC system with a fully-fledged small labelled dataset using the knowledge from other domains (such as audio event detection).

Given the successful application of time-frequency patterns, further research may investigate a unified *time-frequency-spatial* descriptor including also the spatial information. This could be done using multi-channel inputs to CNNs. To date, very little work has addressed the ASC problem using multi-microphone approaches [54, 116] and, that which has a maximum of 2 microphones has considered. In this sense future work should consider multi-microphones or microphone arrays.

In terms of applied research, it is believed that solutions to ASC task require advances in the following areas:

1. open-set protocols and metrics for future public evaluation. In order to realise the commercial potential of ASC and to reduce the gap between fundamental and applied research, the performance of solutions developed in the lab has to be confirmed by users or through *in-the-field* tests;

2. robustness to high-quality as well as low-quality recordings. Invariance to poor recording quality must be further investigated;

3. model complexity in terms of memory and computational constraints. Even while providing good generalisation performance, deep learning solutions may contain millions of parameters. Recent works have been presented for reducing the number of parameters in a CNN model [163] and should be investigated in the context of ASC;

4. *real-time* strategies involving analysing audio in a streaming fashion. Due to limited resources, low-power devices cannot store a huge quantity of audio. This means that real-time ASC systems should use a limited audio sample buffer to extract features and perform predictions. A possible candidate approach may use evolving topologies of neural networks to process raw audio samples directly [164] in a continuous streaming fashion. This may result in a trade-off between flexibility (in terms of number of network parameters) and performance;

5. with the objective of having a reliable acoustic context-recognition system, cues from heterogeneous sensors (camera, motion sensors, temperature sensors) may provide a better *view* of the surrounding environment. Under this assumption, it is clear that the fusion or combination of heterogeneous information sources could be a future axe of investigation.

Finally, the ASC community has grown in the recent years and now attracts interest from both academia and industry. In order to create useful and usable ASC applications, a synergy between fundamental and applied research must become the standard pathway for future research. It is therefore hoped that the analysis presented in this thesis may help steer ASC research in the future.

Part III

APPENDIX

A

APPENDIX

A.1 SUPPORT VECTOR MACHINES FORMULATION

Let us define the *Lagrangian* with several inequality constraints be defined as:

$$L(\boldsymbol{w}, \boldsymbol{\alpha}) = f(\boldsymbol{w}) + \sum_n \alpha_n g_n(\boldsymbol{w}), \tag{46}$$

where f is the optimization function, g_n are the inequality functions and α_n the Lagrangian multipliers. Optimal parameters $\boldsymbol{w}^*, b^*, \boldsymbol{\alpha}^*$ respect the KKT conditions if:

$$\begin{cases} \dfrac{\delta L(\boldsymbol{w}, \boldsymbol{\alpha})}{\delta \boldsymbol{w}_n} = 0 & \forall n \\ \alpha_n \geqslant 0 & \forall n \\ \alpha_n g_n(\boldsymbol{w}) = 0 & \forall n \\ g_n(\boldsymbol{w}) \leqslant 0 & \forall n \end{cases} \tag{47}$$

The following optimization can be easily applied to the 'best boundary' search, in the form $L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_n \alpha_n [y_n(\boldsymbol{w}^T \boldsymbol{x}_n + b) - 1]$. Note that $f(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2$ and the inequality constraints $g(\boldsymbol{w})$ are equal to $[y_n(\boldsymbol{w}^T \boldsymbol{x}_n + b) - 1]$. The routine to transform the primal to a dual problem is described as follows:

Step 1: determine partial derivatives with respect to $\boldsymbol{w}$ and b

$$\begin{cases} \dfrac{\delta L(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\delta \boldsymbol{w}_n} = 0 : & \displaystyle\sum_n \alpha_n y_n \boldsymbol{x}_n = 0 \\ \dfrac{\delta L(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\delta b} = 0 : & \displaystyle\sum_n \alpha_n y_n = 0 \end{cases} \tag{48}$$

Step 2: substitute the partial derivatives into the Lagrangian

$$\begin{aligned} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_n \alpha_n [y_n(\boldsymbol{w}^T \boldsymbol{x}_n + b) - 1] \\ &= \sum_n \alpha_n - \frac{1}{2} \sum_{n,m} \alpha_n \alpha_m y_n y_m \boldsymbol{x}_n^T \boldsymbol{x}_m \end{aligned} \tag{49}$$

Step 3: formulate the dual problem

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \sum_n \alpha_n - \frac{1}{2} \sum_{n,m} \alpha_n \alpha_m y_n y_m \boldsymbol{x}_n \boldsymbol{x}_m \\ \text{s.t.} \sum_n \alpha_n y_n = 0, \quad \alpha_n \geqslant 0 \end{aligned} \tag{50}$$

The best margin is the solution of the dual problem in Eq. 50. The solutions depends not on $\boldsymbol{w}$ but on the Lagrangian multipliers $\boldsymbol{\alpha}^*$. From Eq. 48, $\boldsymbol{w}^* = \sum_n \alpha_n^* y_n x_n$. When a new sample $\boldsymbol{z}$ is classified, the boundary is calculated with respect to these values

$$\boldsymbol{w}^T \boldsymbol{z} + b = (\sum_n \alpha_n y_n x_n)^T \boldsymbol{z} + b = \sum_n \alpha_n y_n \underbrace{\boldsymbol{x}_n \boldsymbol{z}}_{\text{inner product}} + b. \tag{51}$$

Eq.51 depends on the inner product of the new sample z and the samples in the training. KKT equality condition for $\alpha_n g_n(w) = 0$, where $g(\boldsymbol{w}) = y_n(\boldsymbol{w}^T \boldsymbol{x}_n + b) - 1$.

KKT conditions determine the training samples which "support" the final classification. These *special* samples are called SVs. From the KKT conditions, when $\alpha_n > 0$, $g_n(\boldsymbol{w}) = 0$ and so $\boldsymbol{w}^T \boldsymbol{x}_n + b = 1$. The corresponding samples for which $\alpha_n > 0$ lie exactly on the boundary. On the contrary, when $\alpha_n = 0$, $g_n(\boldsymbol{w}) > 0$ so $(\boldsymbol{w}^T \boldsymbol{x}_n + b) > 1$ and the corresponding Lagrangian indicates samples which are not on the boundary.

A.2 T-SNE VISUALISATION FORMULATION

Let $\mathcal{X} = \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N$ be the data set composed of high-dimensional samples and consider that there exists a function which transforms pairwise distances into similarities. Stochastic neighbor embedding (SNE) was reported initially in [165]. It learns a low-dimensional representation of the high-dimensional samples and its goal is to minimize the *difference* between high and low-dimensional representations. The low-dimensional samples are expressed as $\mathcal{Y} = \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N$. The pairwise similarity between samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is defined as a conditional probability $P_i = \Pr(\boldsymbol{x}_j | \boldsymbol{x}_i)$, that $\boldsymbol{x}_i$ would pick $\boldsymbol{x}_j$ as its neighbour according to a Gaussian distribution centred in $\boldsymbol{x}_i$:

$$\Pr(\boldsymbol{x}_j | \boldsymbol{x}_i) = \frac{\exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 / 2\sigma^2)}{\sum_k^N \sum_{l \neq k}^N \exp(-\|\boldsymbol{x}_k - \boldsymbol{x}_l\|^2 / 2\sigma^2)}, \tag{52}$$

where σ is the variance of the Gaussian distribution and N is the total number of samples.

In practice, the computation of all the pairwise distances (expressed in the denominator of Eq. 52) is expensive. Because of that, the conditional probability $\Pr(\boldsymbol{x}_j | \boldsymbol{x}_i)$ replaces the normalization term with a *local normalization* with respect to the neighbourhood K of the considered sample $\boldsymbol{x}_i$:

$$\Pr(\boldsymbol{x}_j | \boldsymbol{x}_i) = \frac{\exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i}^K \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_k\|^2 / 2\sigma_i^2)}, \tag{53}$$

where σ_i is the variance of the Gaussian distribution centred in $\boldsymbol{x}_i$. The variance is scaled for each i^{th} sample such that the number of the considered neighbours is fixed to a parameter (called *perplexity*). Different parts of the space may have different densities of samples. The use of an *adaptive* σ_i allows the algorithm to better adapt to different densities: for dense areas of the space, a small value of σ_i is preferable than sparser areas.

In order to minimize the difference of the two representations, a Kullback-Leibler (KL) divergence criteria is used to quantify the mismatch between probability distributions in the high and low dimensional space. The cost function J minimizes the sum of KL divergences on the overall conditional probabilities using a gradient descent approach.

The low-dimensional representation changes depending on the mismatch between the two probability distributions:

$$J = \sum_{i}^{N} KL(P_i||Q_i) = \sum_{i}^{N}\sum_{j}^{N} Pr(\boldsymbol{x}_j|\boldsymbol{x}_i) \ln \frac{Pr(\boldsymbol{x}_j|\boldsymbol{x}_i)}{Pr(\boldsymbol{y}_j|\boldsymbol{y}_i)}, \tag{54}$$

where P_i indicates the conditional probability distribution for all samples with respect to $\boldsymbol{x}_i$ and Q_i stands for the conditional probability for all low-dimensional samples given $\boldsymbol{y}_i$.

t-SNE is the evolution of SNE and it is different in two aspects: i) it uses a symmetrised version of cost function J to reduce the number of possible combinations; ii) it adopts a Student t-distribution to compute the similarity in low-dimensional space.

Another possible way to represent the cost function J is to use KL divergence between a joint probability P and Q instead

$$J = KL(P||Q) = \sum_{i}^{N}\sum_{j}^{N} Pr(\boldsymbol{x}_i, \boldsymbol{x}_j) \ln \frac{Pr(\boldsymbol{x}_i, \boldsymbol{x}_j)}{Pr(\boldsymbol{y}_i, \boldsymbol{y}_j)}. \tag{55}$$

This is called symmetric SNE because $Pr(\boldsymbol{x}_i, \boldsymbol{x}_j) = Pr(\boldsymbol{x}_j, \boldsymbol{x}_i)$ and $Pr(\boldsymbol{y}_i, \boldsymbol{y}_j) = Pr(\boldsymbol{y}_j, \boldsymbol{y}_i)$. The most relevant advantage of this version is to produce a simpler and more efficient gradient [77].

The low-dimensional similarities between corresponding samples $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are instead given by

$$Pr(\boldsymbol{y}_i, \boldsymbol{y}_j) = \frac{(1 + ||\boldsymbol{y}_i - \boldsymbol{y}_j||^2)^{-1}}{\sum_{k}^{N}\sum_{l\neq k}^{N} \exp(-||\boldsymbol{y}_k - \boldsymbol{y}_l||^2)^{-1}}. \tag{56}$$

The application of the heavy-tailed Student t-distribution in Eq.56 with one degree of freedom allows the modelling of moderate distances in the high-dimensional space by much larger distances in the embedding. This represents more accurately close samples in the high-dimensional space with small distances in the low-dimensional space. Without this distribution, small distances in the the low-dimensional space will just collapse toward specific regions of the space, making any interpretation impossible.

FRENCH VERSION

CLASSIFICATION DES SCÈNES ACOUSTIQUES

INTRODUCTION

Imaginez fermer vos yeux pendant un moment et écouter attentivement les sons dans votre environnement immédiat. Vous pouvez reconnaître des sons spécifiques comme des pas, la climatisation, le passage de voitures ou des voix. Même en l'absence de repères visuels, les humains peuvent identifier la plupart des événements et des sons avec des signaux acoustiques. Ces signaux acoustiques fournissent des informations sur les objets qui ne sont pas dans le champ de vision de l'auditeur. La recherche présentée dans cette thèse porte sur la reconnaissance d'une scène acoustique spécifique par des machines.

Le choix des signaux acoustiques pour reconnaître l'environnement environnant est motivé par l'omniprésence du microphone dans les smartphones, des appareils avec la sphère de l'internet des objets, des wearables et des appareils auditifs. Alors que certains appareils sont équipés par plusieurs capteurs hétérogènes (par exemple des capteurs de lumière, des gyroscopes et des accéléromètres), les capteurs acoustiques sont les plus largement utilisés dans la pratique. Il existe des preuves [1] que la reconnaissance du contexte utilisant des signaux acoustiques donne de meilleures performances que l'utilisation de mesures accélérométriques seulement. Dans tous les cas, les signaux acoustiques et autres sont complémentaires dans un cadre de fusion des sensors.

La classification des scènes acoustiques (CSA) vise à classer l'environnement dans lequel un appareil est utilisé. Le problème de la reconnaissance des scènes acoustiques est particulièrement pertinent dans le cas des appareils mobiles compte tenu de leur utilisation dans des situations multiples au cours d'une journée type. Ici, par exemple, le volume de la sonnerie d'un portable peut être ajusté selon que l'utilisateur est dans un bus, au bureau ou à la maison. La motivation de ce travail provient de la demande continue de fonctionnalités avancées en adaptant automatiquement la configuration de l'appareil à la situation ou au contexte. De plus, le caractère industriel de ce doctorat a conditionné les pistes et les axes de recherche. L'CSA étant un domaine d'étude récent, il existe toujours un écart entre le monde universitaire et l'industrie en termes de problèmes, de solutions, de protocoles et de paramètres; il existe des différences claires entre l'évaluation en laboratoire et la performance sur le terrain. Cette dichotomie explique la structuration de cette thèse en deux parties; une liée à la recherche fondamentale; l'autre lié à la recherche appliquée. L'objectif final est de concevoir un système CSA robuste qui analyse et classe les scènes acoustiques en temps réel sur des appareils de faible puissance.

Applications du CSA

Les applications qui peuvent bénéficier directement de l'CSA englobent les technologies existantes, des smartphones aux aides auditives:

Les dispositifs de sensibilisation au contexte comprennent des capacités d'écoute permanente pour adapter le comportement à la situation environnante [13]. Les exemples incluent l'adaptation d'un volume de sonnerie selon que l'utilisateur se trouve dans un bus, dans un bureau ou au cinéma [14]. Les preuves [15] montrent que la capacité d'associer un comportement à un contexte est particulièrement pratique pour les utilisateurs. Un autre exemple d'applications pratiques est rapporté dans [16], où les dispositifs portables ajustent

le taux (ou l'intensité) des notifications en fonction du contexte. Le coût d'être distrait par un appareil peut être élevé: imaginez recevoir de nombreuses notifications dans la voiture en conduisant, au restaurant avec d'autres personnes ou en traversant la rue. La décision de notifier ou non et comment notifier l'utilisateur doit être prise en considération pour le contexte actuel.

Les robots d'écoute utilisent l'information de «où je suis» pour changer de comportement. Particulièrement dans des conditions de mobilité élevée, l'information préalable sur la localisation du robot aide à définir les actions les plus appropriées à réaliser [17]. Des exemples concrets peuvent utiliser CSA pour modifier la vitesse du robot, qu'il soit situé à l'intérieur ou à l'extérieur [18].

Le marquage automatique des données exploite les similitudes de contexte pour l'étiquetage automatique des données audiovisuelles. Il existe une énorme quantité de contenu multimédia non segmenté, ni étiqueté, dont le marquage manuel serait pratiquement impossible. La combinaison de la vidéo, de l'image et de l'information sur la scène acoustique permettrait de marquer automatiquement une grande quantité de matériel. Ce matériel pourrait ensuite être utilisé pour recycler CSA avec des ensembles de données plus volumineux [19].

Les aides auditives adaptent leur configuration à l'environnement de l'utilisateur, tel qu'un bureau silencieux, un restaurant ou un music-hall. Les solutions actuelles d'aides auditives sont réglées en fonction d'environnements acoustiques généraux qui ne s'adaptent pas rapidement aux changements de contexte [20]. Les solutions CSA peuvent être utilisées pour améliorer la qualité audio et permettre des configurations basées sur le contexte. Dans toutes les applications ci-dessus, CSA est essentiellement une étape de prétraitement qui fournit des informations préalables à d'autres systèmes.

Contributions

La structure de la thèse reflète la nature des contributions concernant la recherche fondamentale et appliquée. Le plan est illustré graphiquement par une carte mentale dans la Fig. 1. La recherche fondamentale est l'objet de la partie 1 (à gauche de la figure 1) qui décrit les contributions entre le premier défi public sur CSA en 2013 [6] et le deuxième en 2016 [26]. La séquence des chapitres suit temporellement ces deux jalons, relatant les défis DCASE publics en 2013 et 2016. La recherche appliquée est au centre de la partie 2 (à droite de la figure 1) qui traite des implications pratiques de l'CSA dans le monde réel scénarios. Les contributions de cette partie incluent l'adaptation des solutions CSA pour travailler en mode streaming avec une complexité réduite. Le travail rapporté dans cette thèse a donné lieu à plusieurs publications:

- publication 1 (conference paper): "Acoustic context recognition for mobile devices using a reduced complexity SVM", 2015 IEEE European Signal Processing Conference (EUSIPCO);
- publication 2 (conference paper): "Acoustic context recognition using local binary pattern codebooks", 2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA);
- publication 3 (workshop paper): "Acoustic scene classification using convolutional neural networks", 2016 IEEE Detection and Classification of Acoustic Scenes and Events challenge (DCASE);
- publication 4 (conference paper): "The open-set problem in acoustic scene classification", 2016 IEEE Workshop on Acoustic Signal Enhancement (IWAENC);

- publication 5 (conference paper): "Baby cry sound detection: a comparison of hand crafted features and deep learning approach", 2017 Springer Engineering Applications of Neural Networks conference (EANN);
- publication 6 (patent): "Acoustic Context Recognition using Local Binary Pattern Method and Apparatus", US Patent App. 15/141,942
- publication 7 (patent): "Embedded car detector based on acoustic sensor", EU patent App. under approval.

La partie 1 commence par le chapitre 2 qui décrit l'état de l'art de l'CSA en 2013, lors du premier concours public en CSA. Conjugué à un défi public, un ensemble de données a également été publié. Bien qu'étant un grand pas vers la standardisation de la tâche CSA (données, protocoles, métriques d'évaluation), les méthodes standard étaient toujours basées sur des fonctionnalités principalement conçues pour la parole ou la musique (par exemple MFCC). Le système gagnant de ce défi, en fait, estime et modélise les modèles récurrents dans les MFCC. Ce système et ses principales limites sont discutés au chapitre 3, où une première référence est également présentée. Les moyens possibles d'évaluer et de visualiser les fonctionnalités audio sont présentés dans le chapitre 4 qui conduit à la conception de nouvelles fonctionnalités. À ce jour, presque toutes les approches CSA existantes sont basées sur des caractéristiques traditionnelles conçues pour d'autres domaines. Même ainsi, les expériences montrent que ces caractéristiques peuvent ne pas être suffisamment discriminantes pour la tâche CSA.

Compte tenu de la focalisation sur les caractéristiques CSA, la structure acoustique complexe d'une scène est représentée par des motifs spectro-temporels locaux, extraits directement du spectrogramme (publications 2 et 6). Par conséquent, l'idée d'extraire des modèles spectro-temporels est ensuite exploitée en utilisant une topologie particulière des réseaux neuronaux profonds comme indiqué au chapitre 6. Cette contribution (publication 3) a été soumise et évaluée publiquement dans le contexte de l'évaluation DCASE 2016 dont les principaux résultats les tendances sont présentées au chapitre 7.

Les grandes lignes de la partie 2 sont résumées comme suit: Le chapitre 8 décrit les problèmes pratiques de l'CSA. L'ensemble de données NXP, bien que propriétaire, est considéré comme une contribution dans le contexte d'un doctorat cifre. Les données contenues dans cet ensemble de données peuvent être utilisées non seulement pour l'CSA, mais aussi pour d'autres tâches connexes (détection d'événement, mélange de la parole avec enregistrement de scène acoustique pour apprendre un modèle plus robuste, etc.). Les contraintes de calcul en termes de complexité et de mémoire sont abordées au chapitre 8 avec une contribution supplémentaire incluant un système CSA à complexité réduite (publication 1).

L'une des plus grandes limites des systèmes CSA actuels concerne son application aux problèmes d'ensembles fermés. En pratique, les applications CSA sont ouvertes dans la nature, où le nombre de classes pendant l'évaluation est illimité. Les contributions comprennent la proposition d'une nouvelle approche de l'évaluation des solutions CSA avec une approche ouverte, comme indiqué au chapitre 9. Cette contribution (publication 4) présente le problème CSA comme une détection de scène acoustique où un petit nombre de scènes connues sont détecté dans un plus grand univers de classes inconnues. Les conclusions du dernier chapitre 10 rassemblent les réflexions et conclusions des recherches fondamentales (Partie 1) et appliquées (Partie 2), et décrivent des idées pour des recherches futures.

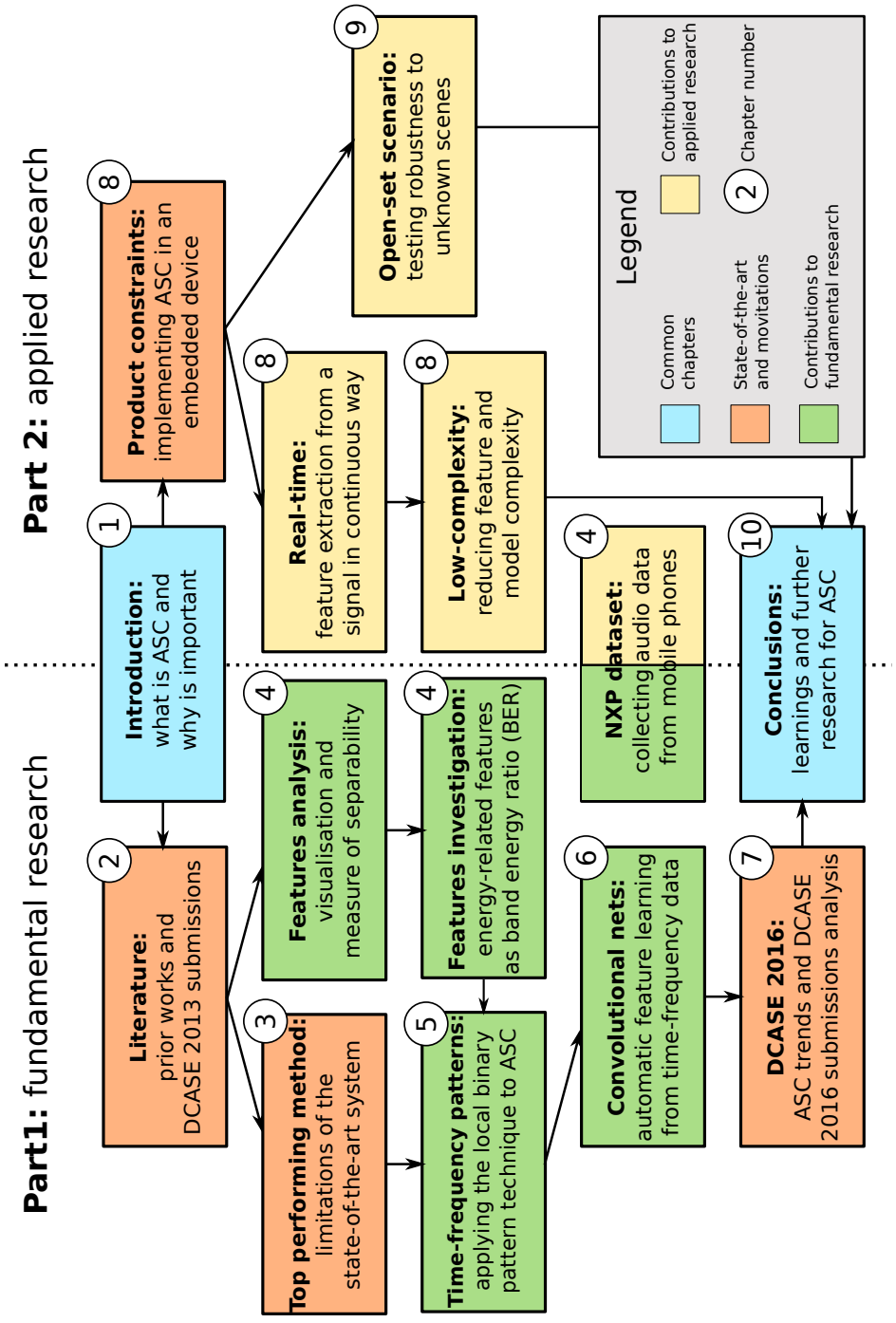


Figure 1: *Mind-map* des blocs principaux composant la thèse. La légende en bas à droite aide à lire l'image entière. Les chiffres dans le coin supérieur droit représentent l'index du chapitre.

Plusieurs approches ont été proposées dans le passé pour classer les sons et les scènes acoustiques, soutenues par des études psycho-acoustiques [5]. L'une des conclusions les plus pertinentes de ces études est que notre système auditif repose sur une mémoire sonore capable d'associer les sons à un environnement significatif. À la lumière de cela, Ellis [34] en 1996 a proposé de décrire une scène acoustique comme un mélange d'éléments de construction plus simples. Dans la même année, Couvreur et al. [34] ont étudié une reconnaissance automatique des sources de bruit ambiant (comme la voiture, le camion, l'avion) en fonction de leurs propriétés acoustiques globales. Cette approche a été développée par El-Maleh et al. [35] en '99 en utilisant des caractéristiques spectrales et un classificateur gaussien. La première méthode traitant spécifiquement du problème de l'CSA concerne un rapport technique de Sawhney et Maes en 1997 [36]. Les auteurs ont enregistré un petit ensemble de données composé de voix de personnes, de métro, de trafic et d'autres classes. À partir de ces enregistrements, ils extraient des caractéristiques basées sur des filtres psycho-acoustiques, en utilisant un classificateur de réseaux neuronaux récurrent. Ils rapportent une précision de classification de 68% sur 5 classes.

Quelques années plus tard en 2001, Peltonen et al. [37] montraient que les humains identifient une scène avec des événements sonores typiques, tels qu'un clic, un claquement de porte ou un moteur de voiture. Les tests effectués sur 19 sujets ont montré une précision globale de classification de 70% sur 25 classes. L'énorme variation des précisions entre les classes (elle varie de 32% à 100%) dépend des indices acoustiques présents dans la scène: lorsque les sons de la scène sont déterminants pour distinguer une classe d'une autre, la précision est plus grande.

Comme prévu, une classification intégrée sur une période plus longue contient des informations plus importantes, comme mentionné précédemment dans [37]. Par conséquent, une longueur idéale pour avoir des résultats de classification stables suggère un signal de 30 à 40 secondes. En dépit de ces observations, l'aspect le plus important de la recherche de Peltonen était d'appliquer pour la première fois le modèle de mélange MFCC et gaussien au problème CSA, atteignant une précision de 68% sur 17 classes. L'adoption de MFCC-GMM a fourni un système de base pour la recherche future. Poursuivant les expériences de Peltonen, Eronen et al. [39] en '03 ont exploité l'évolution temporelle de la scène acoustique pour améliorer le système de base MFCC-GMM, en utilisant un modèle de Markov caché (HMM) à deux états complètement connectés. Ce système a été comparé à la capacité humaine de reconnaître 18 classes et 6 méta-classes (par exemple, extérieur, véhicules, intérieur, etc ...). La précision de la reconnaissance du système HMM est de 61% sur 18 classes contre 69% des tests d'écoute humaine.

Un autre axe de recherche interroge la taxonomie de la scène: quels sont les liens entre l'expérience personnelle quotidienne et l'évaluation collective à travers un concept linguistique de haut niveau? Dubois et al. [40] en '06 ont étudié cette association entre les concepts de haut niveau et les scènes acoustiques. La recherche a montré que les individus classent les scènes acoustiques sur la base d'expériences antérieures. Pour renforcer cette perspective, une étude complémentaire a été menée par Tardieu et al. [41] en '08 sur l'organisation humaine des indices acoustiques dans les niveaux croissants d'abstraction. Dans le contexte d'une scène acoustique d'une gare ferroviaire, ils ont démontré que les gens utilisent des signaux acoustiques locaux (activité humaine) et des informations globales (réverbération, intensité) pour construire hiérarchiquement une scène acoustique. La même idée a été récemment proposée par Torija [42] en '13. En utilisant 15 descripteurs acoustiques, une scène acoustique est composée par ces éléments de construction.

Method (ID)	Features	Classifiers	Testing strategies
Olivetti et al. (OE) [66]	Length of the compressed audio file	Random forest based on the compression distance	
Elizalde et al. (ELF) [57]	MFCCs + Δ + $\Delta\Delta$ over a concatenation of left, right, difference and average of stereo channels	GMM-UBM $\rightarrow$ i-vector	Maximum likelihood
Krijinders et a. (KH) [55]	time-frequency chocleagram	statistics $\rightarrow$ SVM	One-vs-one
Baseline	MFCCs	GMM	Maximum likelihood
Patil et al. (PE) [61]	time-frequency multi-resolution analysis $\rightarrow$ PCA	SVM	One-vs-one, weighted majority vote by the energy present in 1s window (overlap 0.5s)
Nogueira et al. (NR) [54]	MFCCs, temporal features (modulation rate of MFCCs over 4 bands, event density estimation), spatial features (time and amplitude differences between the two channels) $\rightarrow$ Fisher score for features selection	SVM	
Nam et al. (NHL) [62]	unsupervised learning using restricted Boltzman machines on Mel-spectrogram $\rightarrow$ PCA	SVM	One-vs-all
Chum et al. (CHR) [59]	energy/frequency features over short and long frames (different temporal resolutions)	GMM $\rightarrow$ HMM	Maximum likelihood
Geiger et al. (GSR) [53]	spectral, cepstral, energy, voicing-related over 4s of signal	SVM	Majority vote
Rakotoma-monjy et al. (RG) [33]	Histogram of gradients on constant Q transforms	SVM	One-vs-one
Li et al. (LTT) [58]	MFCCs on wavelet decomposition	Ensemble of binary trees	Majority vote
Roma et al. (RNH) [56]	MFCCs $\rightarrow$ recurrent quantification analysis metrics (RQA)	SVM	One-vs-one

Table 2: La liste des systèmes soumis au challenge DCASE 2013, suivie du type de fonctionnalités, du classificateur et des stratégies de test. La flèche exprime les dépendances à partir de la fonctionnalité $\rightarrow$ traitement des entités ou classification.

D'après Räsänen et al. [1] en '11, l'utilisation du classificateur audio combinée à l'accélération a permis d'améliorer les performances de classification du contexte. Au lieu de fusionner des informations sensorielles de bas niveau (c'est-à-dire de combiner directement des caractéristiques provenant de capteurs acoustiques et d'accélération), seules les prédictions de classification sont combinées. En fait, la prédiction finale est une somme pondérée de prédictions uniques provenant de classificateurs acoustiques et d'accélération. Une intuition similaire a été adoptée pour la fusion des indices visuels et acoustiques par Lee et al. [44] en '12. Une approche hiérarchique complète a été proposée dans Feki at al. [45] en '11. Dans cette approche descendante, chaque diffusion audio a été classée en sons vocaux, musicaux ou environnementaux. Si le streaming audio ne contenait pas de paroles ou de musique, il a été classé en fonction de la scène acoustique la plus probable. Cette approche décompose un problème de classification global en tâches de sous-classification plus simples, depuis des concepts de haut niveau jusqu'à des événements sonores uniques.

En termes de reproductibilité et de comparabilité des résultats, le domaine CSA manquait d'un ensemble de données commun. Avant 2013, chaque travail mentionné ci-dessus utilisait un ensemble de données différent (avec un nombre différent de classes et de conditions d'enregistrement). Le premier ensemble de données sur DCASE a été publié en 2013, associé à une évaluation publique des méthodes CSA. Les travaux antérieurs à DCASE 2013 étaient généralement réalisés avec des données variables (la qualité du microphone, les types et le nombre de classes en sont quelques exemples). En conséquence, la plupart des œuvres ont été évaluées en utilisant différentes bases de données d'enregistrements. Le jeu de données de défi DCASE, dont l'objectif principal était de soutenir la reproductibilité et les comparaisons avec d'autres solutions, a abordé exactement ce problème.

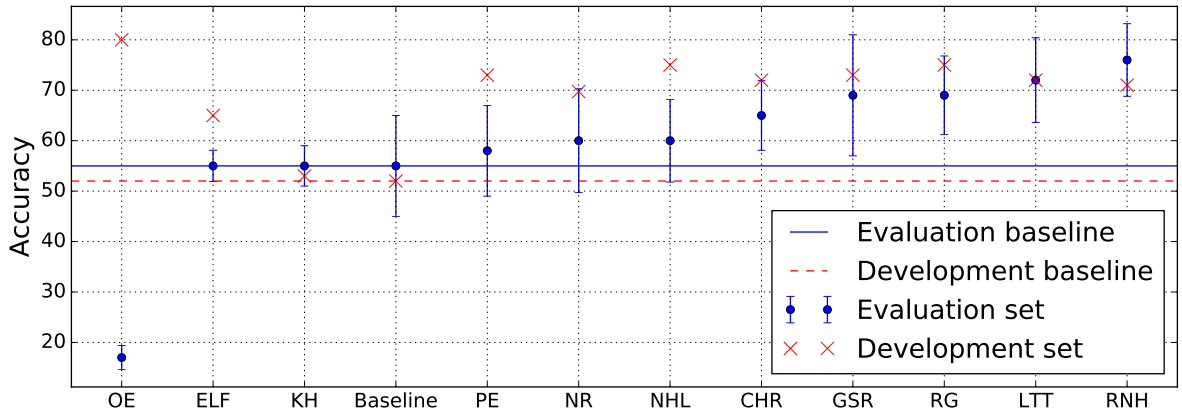


Figure 2: La courbe montre la précision moyenne avec des intervalles de confiance (IC) de 95% sur une validation croisée de 5 pour l'ensemble de données DCASE 2013. Dans les cercles bleus, les valeurs de l'ensemble d'évaluation, dont la ligne de base est également exprimée par une ligne bleue; dans les étoiles rouges les valeurs du développement avec la ligne de base exprimée en ligne rouge pointillée. Pour certains systèmes, les IC ne sont pas fournis dans la description de l'ensemble de développement et n'ont pas été signalés.

DCASE 2013 résultats

Le tableau 2 montre une différence significative entre les résultats de l'évaluation et les performances de l'ensemble de développement. L'abréviation des travaux soumis est reportée dans la figure 2. Certaines méthodes étaient probablement surimposées aux données de développement. En général, les meilleurs systèmes (LTT, RNH) ont amélioré les performances dans la phase d'évaluation. Un nombre important de systèmes fonctionnent mieux que la ligne de base et même dans le cas d'une précision similaire, les systèmes ELF ou KH devraient être préférés pour un intervalle de confidence inférieure.

Par conséquent, les performances et les méthodes sont fortement corrélées. À l'exception des systèmes ELF et CHR, tous les autres utilisent des classificateurs discriminatifs (SVM, arbre binaire). Sur le plan des caractéristiques, les MFCC sont les plus adoptés. Parmi plusieurs stratégies de test, le vote majoritaire semble être le plus efficace permettant d'intégrer les décisions dans le temps. Ceci suggère qu'une scène acoustique est détectée de manière fiable à 30s, comme on le trouve dans [38].

En raison de sa large adoption par de nombreux systèmes soumis, SVM ne fait pas la différence en termes de performance finale. En effet, en analysant les trois meilleurs systèmes, les deux systèmes RG et RNH proposent des fonctionnalités adaptées à CSA: la première en capturant des structures temporelles en utilisant une représentation d'images basée sur CQT; le second en quantifiant la récurrence des MFCC consécutifs. L'idée d'exploiter des spectrogrammes temps-fréquence est commune à d'autres systèmes (KH, PE, NHL, LTT) suggérant que l'information temporelle est pertinente pour la tâche CSA.

D'un point de vue global, le fait que seulement quelques systèmes surpassent la base prouve la difficulté de la tâche pour une quantité modeste de données. De plus, il semble qu'un niveau de performance similaire obtenu dans d'autres domaines (tels que la reconnaissance de la parole ou la classification des genres musicaux) pourrait être atteint à partir d'une enquête plus approfondie sur les caractéristiques adaptées à CSA.

MODÈLES TEMPS-FRÉQUENCE

Presque toutes les approches de l'CSA utilisent des caractéristiques traditionnelles conçues principalement pour les applications de traitement de la parole telles que la reconnaissance

de la parole ou du locuteur. Même ainsi, les expériences des chapitres précédents ont montré que ces caractéristiques peuvent ne pas être suffisamment discriminantes pour la tâche CSA. Voici les principaux inconvénients des systèmes CSA actuels:

1. ils ne capturent pas les informations globales et locales. Les caractéristiques déterminent si un système représente une information de scène générique (telle que l'énergie globale, l'enveloppe spectrale, etc.) ou s'il décrit une variation relative locale (comme BER, RMS). L'utilisation de l'information mondiale et locale s'est avérée efficace pour la littérature d'CSA [34, 45], même s'il n'existe pas d'approche globale;

2. ils sont basés sur des fonctionnalités non adaptées à CSA. Par exemple, les MFCC restent le choix standard dans de nombreux systèmes CSA. Les MFCC ne capturent que les variations à court terme avec une information dynamique minimale, tandis que la corrélation dans le domaine temporel peut aider à différencier les différentes scènes. A titre d'exemple, une approche prometteuse [33] représente une structure acoustique complexe avec des caractéristiques à la fois dans l'espace temps et dans l'espace fréquence. Intuitivement, les caractéristiques spectro-temporelles devraient être considérées comme une alternative aux approches standard basées sur le MFCC;

3. ils impliquent une structure temporelle même en présence d'une séquence de sons clairsemée et non ordonnée. Contrairement aux signaux de parole, où une structure temporelle forte est déterminée par la séquence téléphonique, CSA est caractérisée par une structure temporelle relativement faible. Les événements composant une scène peuvent survenir à n'importe quel moment et dans n'importe quel ordre et durée. Comme nous l'avons montré dans [37], les auditeurs humains classent une scène par la présence d'un son particulier. Cela suggère que se concentrer sur la présence de certains sons peut améliorer les performances, comme indiqué dans [12].

Local binary patterns (LBP)

L'idée originale de LBP est décrite dans [93]: l'opérateur représente des images texturales complexes d'une manière simple et pratique à travers le seuillage binaire des voisins environnants de chaque pixel. Chaque bloc autour d'un pixel fournit un nombre binaire qui exprime les relations des pixels par rapport au pixel central: si la différence des voisins et du pixel central est négative, le résultat est 0 sinon il est 1. Un histogramme h représente la fréquence des nombres binaires dans chaque bloc. L'histogramme lui-même exprime l'image (ou une partie de celle-ci) comme les occurrences de motifs binaires trouvés dans l'image. L'application de LBP aux spectrogrammes nécessite une certaine adaptation. Chaque bac du spectrogramme reflète la quantité d'énergie présente à proximité des intervalles de temps et de fréquence spécifiques. Les spectrogrammes, par construction, sont caractérisés par des fluctuations locales de la poubelle (à savoir des poubelles qui peuvent varier de manière significative dans une zone locale), ce qui peut dégrader la représentation des caractéristiques LBP. LBP est fortement affectée par les fluctuations des bacs dans le voisinage qui peuvent en effet changer radicalement le code binaire LBP. Dans l'analyse LBP, ces fluctuations sont des transitions rapides dans un code LBP de 1 à 0 et vice-versa.

Ainsi, l'interpolation des valeurs de bin aide à atténuer l'effet de ces fluctuations en lissant globalement les blocs (figure 3). Une autre stratégie pour ajouter de la robustesse à LBP est de considérer uniquement les codes LBP pour lesquels le nombre de transitions entre 0 et 1 est inférieur ou égal à 2. Ce sous-ensemble de LBP représente les modèles dits uniformes. Les motifs non uniformes restants sont souvent regroupés et considérés comme un motif non uniforme unique e distinct.

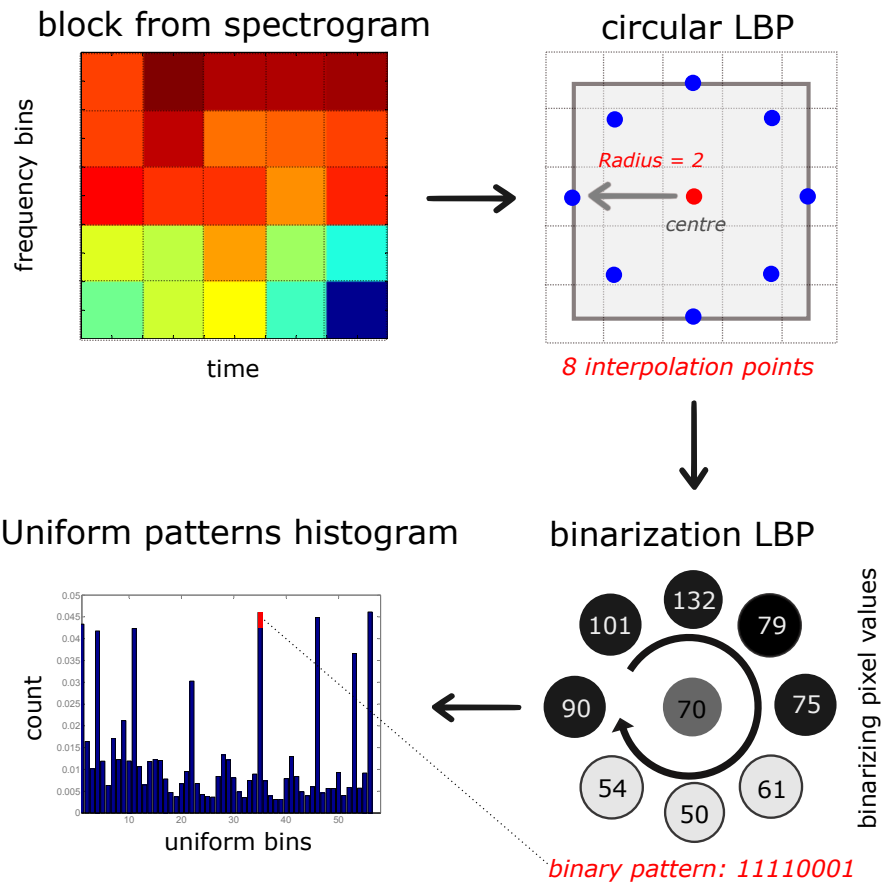


Figure 3: Du bloc spectrogramme à l'histogramme LBP: à partir du coin supérieur gauche de l'image, le bloc spectrogramme est analysé en utilisant LPB8,2 avec 8 voisins et rayon égal à 2; le code binaire local est ensuite généré; enfin le code binaire est mis à jour dans la case correspondante de l'histogramme.

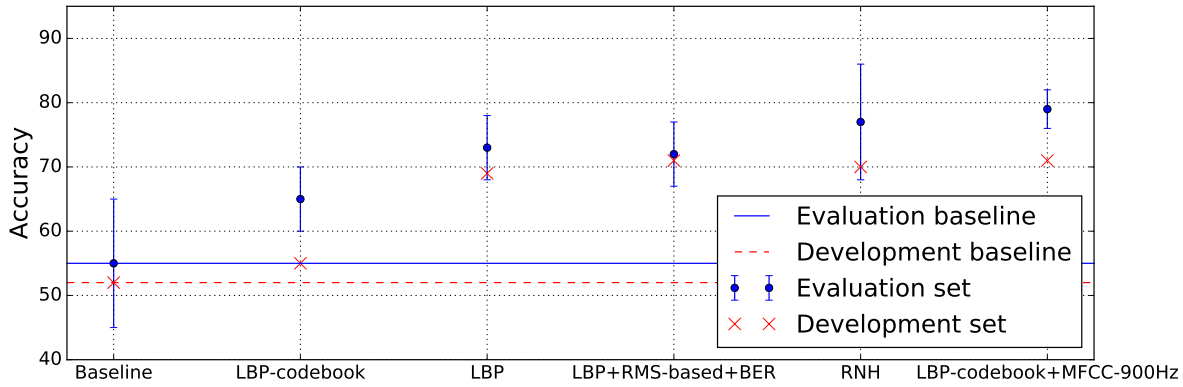


Figure 4: La courbe montre la précision moyenne avec des intervalles de confiance (IC) de 95% sur une validation croisée de 5 pour l'ensemble de données DCASE 2013. Dans les cercles bleus, les valeurs de l'ensemble d'évaluation, dont la ligne de base est également exprimée par une ligne bleue; dans les étoiles rouges, les valeurs de l'ensemble de développement avec la ligne de base exprimée en ligne rouge pointillée. A l'exception de la ligne de base et de la RNH, les autres systèmes ont été proposés dans ce travail.

Les systèmes LBP proposés sont comparés aux systèmes de pointe de la Fig. 5. Les LBP atteignent une précision de 73%, soit 18% de mieux que la base de référence MFCC. En outre, remplacer RQA dans le système RNH donne une précision de 79%. La LBP combinée à des caractéristiques basées sur l'énergie (BER et RMS) atteint une précision de 72%.

À première vue, les fonctionnalités basées sur le protocole LBP surpassent les fonctionnalités MFCC-noCo pour tous les jeux de données. L'ajout de fonctionnalités RMS et BER à LBP augmente encore les performances, atteignant la meilleure précision pour les jeux de données DCASE 2013 dev, NXP et Rouen. Pour les ensembles de données plus volumineux tels que NXP et Rouen, la configuration LBP + RMS-BASED + BER atteint respectivement 93% et 88%.

En particulier, pour l'ensemble de données de Rouen, les résultats sont comparables à la précision de 87% rapportée dans [33], qui utilise des techniques de traitement d'image appliquées à une représentation temps-fréquence. Le système MFCC-RQA-900 reste le deuxième meilleur système pour l'ensemble de données DCASE 2013 (eval) mais il se généralise mal à d'autres ensembles de données. Étonnamment, pour les ensembles de données NXP et Rouen, l'ajout de RQA aux entités MFCC n'a aucun impact. Le système de livre de codes LBP atteint une précision de 90% pour l'ensemble de données NXP, tandis que lorsqu'il est combiné avec MFCC, il améliore encore les performances, atteignant la précision la plus élevée atteinte pour l'ensemble d'évaluation DCASE 2013. Enfin, le système BER + RMS + RMS semble être le plus cohérent parmi les quatre ensembles de données.

Convolutional neural network

La littérature de l'CSA montre que la majorité des approches CSA utilisent des fonctionnalités développées pour d'autres tâches connexes telles que la reconnaissance de la parole ou de la musique (revue de la littérature au chapitre 2). Des travaux récents ont exploré l'utilisation de caractéristiques qui capturent la corrélation temps-fréquence. Certains de ces travaux s'appuient sur des méthodes populaires dans d'autres domaines bidimensionnels tels que le traitement d'image. LBP, par exemple, représente un spectrogramme audio avec un histogramme des motifs les plus fréquents [98]. De même, HOG code la direction des variations dans les spectrogrammes à base de CQT [33]. Après avoir été appliquées avec succès à d'autres problèmes connexes, des techniques d'apprentissage en profondeur [6] sont en train d'émerger [99]. Les réseaux neuronaux profonds (DNN)

sont capables d'identifier et d'extraire des caractéristiques discriminantes optimisées à partir des données d'apprentissage et offrent ainsi une alternative aux fonctionnalités artisanales. De nombreuses architectures et représentations d'entrée de données ont été étudiées pour une multitude d'applications différentes telles que la reconnaissance d'image et de reconnaissance vocale [100, 101].

Alors que la première étude des approches DNN de l'CSA [99] a montré des résultats prometteurs, le travail était basé sur les caractéristiques du MFCC. Ainsi, le bénéfice potentiel de l'apprentissage en profondeur était encore limité par l'utilisation initiale des MFCC. Cette partie présente les travaux expérimentaux avec une approche particulière de l'apprentissage en profondeur impliquant des réseaux de neurones convolutionnels (CNN). Les principales raisons de ce choix sont (i) la possibilité de remplacer les caractéristiques artisanales par des caractéristiques apprises automatiquement et (ii) la possibilité d'utiliser des représentations temps-fréquence comme entrées du réseau, en accord avec les recherches antérieures sur LBP spectro-temporelle.

Les réseaux CNN ont une architecture de réseau profonde et multicouche. Différemment des MFCC qui décorrélent les données avec le DCT, CNN prend en entrée le spectrogramme log-mel filtré imitant un comportement de traitement d'image. Dans la couche convolutionnelle, chaque unité cachée n'est pas connectée à toutes les entrées de la couche précédente, mais uniquement à une zone de l'espace d'entrée original, appelée champ réceptif. Ces petites parties de tout l'espace d'entrée sont connectées aux unités cachées à travers les poids w et bias b. Cette opération est équivalente à un traitement de filtre convolutif. L'architecture proposée dans ce travail est illustrée à la Fig. 6. Elle est composée d'une couche d'entrée, d'une pile de couches convolutionelles et de regroupement, d'une couche cachée entièrement connectée et d'une couche de sortie finale. Les CNN s'appuient sur des opérations de convolution et de regroupement: la couche convolutionnelle applique un ensemble de filtres sur une partie de l'entrée dont les filtres sont partagés dans tout l'espace d'entrée; la mise en commun peut être considérée comme une opération de sous-échantillonnage qui se concentre davantage sur le modèle lui-même que sur l'emplacement exact dans l'entrée. Cela ajoute de la robustesse aux petites modifications et traductions dans l'espace d'entrée. Une architecture profonde réplique ces opérations dans une pile. De cette manière, les filtres de chaque couche capturent des motifs à un niveau d'abstraction plus élevé, car ils travaillent sur des entrées de résolution inférieure provenant de la couche de regroupement. Finalement, la couche entièrement connectée connecte les unités provenant de toutes les positions locales pour effectuer une classification globale de l'entrée. Comme pour les histogrammes LBP, les entrées du spectrogramme initial sont représentées par la combinaison de leurs composantes locales.

Dans la section suivante, un examen plus détaillé des meilleures méthodes est présenté. Les noms des soumissions sont les mêmes que ceux utilisés dans les résultats DCASE 2016 de la figure 6. Le premier système Battaglino_1 proposé a atteint une précision de 80% en utilisant deux couches convolutives sans normalisation de lot. Le deuxième système Battaglino_2 adopte la normalisation par lots et une forme de filtre carré 5×5. Avec une précision de 5% inférieure à celle du meilleur système (89,7%), l'architecture profonde proposée est toujours capable de surpasser un système MFCC-GMM standard grâce à l'apprentissage automatique de fonctions significatives.

CSA comme problème de classification ouvert

Le problème de la classification en CSA a été vu jusqu'à présent comme celui d'attribuer à une scène acoustique une étiquette qui correspond à un ensemble fermé de classes. Si le classificateur ne connaît que deux sorties dans l'ensemble d'apprentissage (par exemple

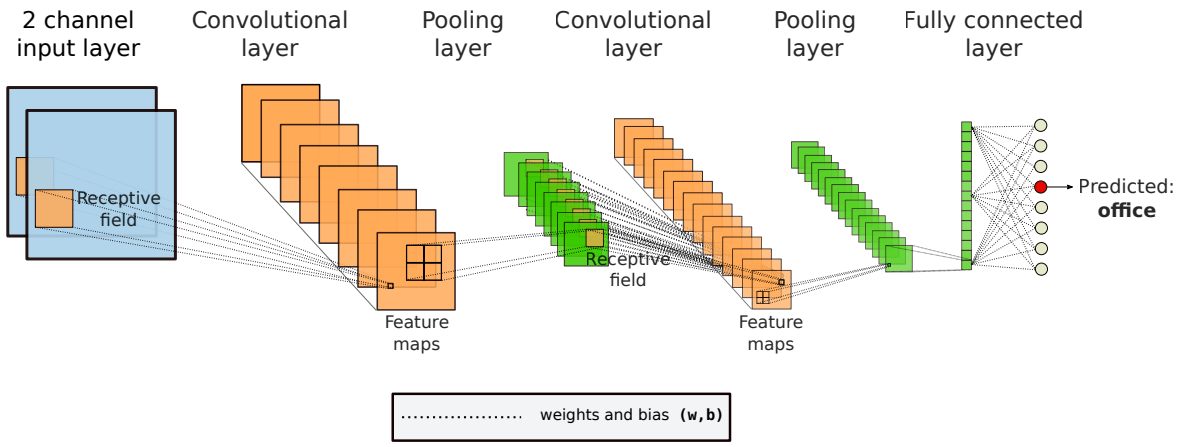


Figure 5: Un exemple d'architecture CNN étudiée dans ce travail: l'entrée est un spectrogramme statique et dynamique à 2 canaux. Ils sont suivis de deux couches de convolution et de regroupement empilées. Les couches entièrement connectées et en sortie produisent les probabilités des données d'entrée appartenant à chaque classe acoustique.

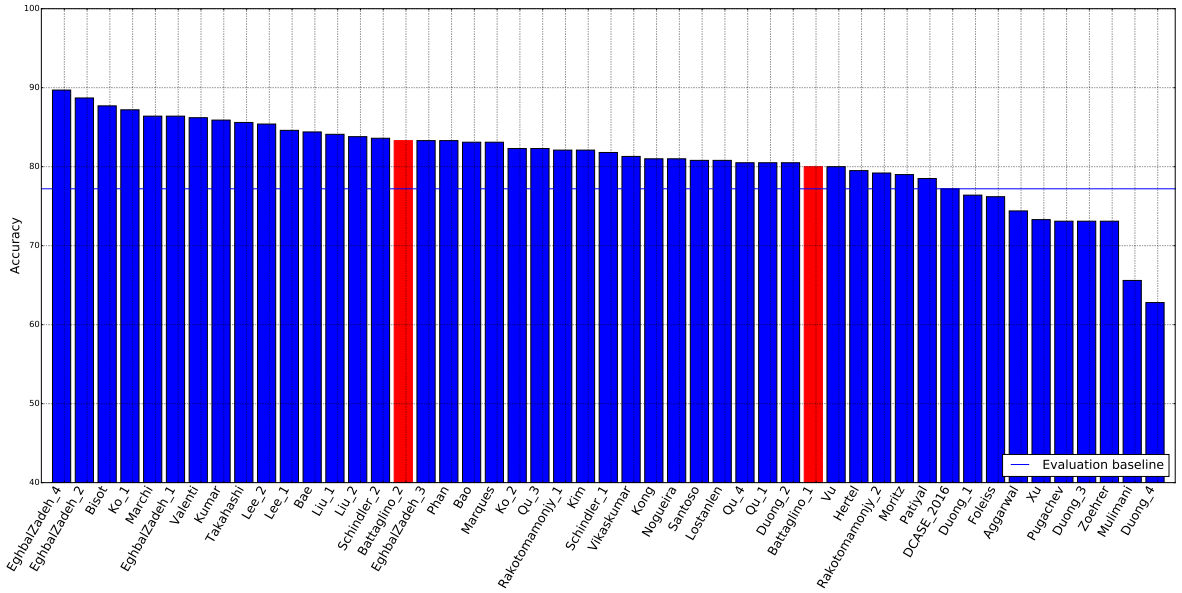


Figure 6: Résultats sur l'ensemble d'évaluation DCASE 2016. Le système de référence a une précision globale de 77,2% et il est indiqué par une ligne bleue continue. Le nom du système suit la même dénomination des soumissions de défi. En rouge continu, les systèmes basés sur CNN Battaglino_1 et Battaglino_2.

voiture et bureau), il classera toutes les autres scènes comme l'une des deux, même si la scène ne correspond ni à une voiture ni à un bureau (par exemple un train). Du point de vue de l'application, cette approche fera des affectations ou des décisions sans signification: exemples sont des applications qui basculent automatiquement la sonnerie en mode silencieux dans le bureau pourrait se déclencher aussi bien dans le parc ou dans d'autres environnements qui ne sont pas présents dans le ensemble fermé de classes.

Commun à tous les travaux passés, est l'évaluation des systèmes CSA dans un scénario fermé pour lequel des données d'entraînement sont disponibles pour chaque classe acoustique qui peut être rencontrée pendant les essais. Cette stratégie d'évaluation ne reflète pas les applications pratiques dans lesquelles des données hors-jeu peuvent être facilement rencontrées. Sans possibilité de rejeter les données acoustiques hors classe, leur affectation à une classe cible entraînera une dégradation des performances de classification. En tant que telles, les approches actuelles de l'évaluation des systèmes CSA ne définissent pas le niveau de performance auquel on peut s'attendre dans la plupart des applications pratiques. De manière surprenante, aucun travail antérieur n'a étudié l'CSA dans un scénario ouvert.

Le concept de l'ouverture

Bien que le concept de problèmes ouvert fermé et est maintenant clairement défini, la nécessité d'évaluer la performance de l'CSA dans un scénario ensemble ouvert conduit à un concept relatif d'ouverture. Un système CSA est conçu pour classer un certain nombre de classes cibles. En plus des classes cibles, il existe un certain nombre de classes négatives connues. Tout échantillon de données qui ne se trouve dans aucune de ces deux classes est désigné comme un membre de la classe inconnue. Formellement, une évaluation en circuit ouvert implique donc une combinaison de t classes cibles, k classes négatives connues et u classes négatives inconnues. Leurs valeurs sont définies selon un scénario d'évaluation ou un protocole comme suit: un ensemble de données d'apprentissage est composé de données des classes t et k alors qu'un ensemble de données de test combine des données de classes connues t et k avec des données supplémentaires de classes inconnues.

Le besoin d'évaluation et le scénario particulier imposent certaines contraintes sur les valeurs de t, k et u. Tandis que u est, par sa définition même, illimitée, l'évaluation des systèmes CSA peut nécessiter la définition d'un nombre théoriquement fini de classes inconnues; la valeur de t, k et u peut refléter la difficulté d'une évaluation. Les tâches impliquant de plus grandes valeurs de u et k par rapport à t sont comparativement plus difficiles que les tâches avec des valeurs plus petites. En particulier, les classes négatives inconnues sont comparativement plus difficiles à gérer que les classes négatives connues. Un travail connexe [138] définit une mesure, appelée «ouverture», qui reflète la difficulté d'une telle tâche de classification. En s'appuyant sur le travail original susmentionné, une mesure d'ouverture est ici exprimée en termes de t, k et u comme:

$$\text{openness} = 1 - \sqrt{\frac{t+k}{t+k+u}}. \tag{57}$$

Une ouverture de 0 induit un problème d'ensemble fermé, alors qu'une ouverture de 1 est un problème entièrement ouvert. La racine carrée tempère les augmentations rapides de l'ouverture avec seulement u modéré. Étant donné un nombre fixe de cibles t, le niveau d'ouverture dépend de k et u: k, le niveau d'ouverture tendra à 1; quand u 0 le niveau d'ouverture tendra vers 0. Selon cette hypothèse, la valeur d'ouverture concerne u, le nombre de classes inconnues présentées lors des tests.

Alors que les ensembles de données publiquement disponibles pour l'CSA n'empêchent pas une évaluation en ouvert, les protocoles d'évaluation standard sont tous fermés (u = 0).

Un classificateur adapté à l'open-set

Une approche particulière, appelée description de données vectorielles de support (SVDD), apprend une hypersphère dans laquelle des échantillons cibles sont contenus [145]. Le but est de représenter les données cibles dans le plus petit volume d'hypersphère possible. En utilisant les données cibles uniquement à des fins de formation, le SVDD évite de surenchérir sur les négatifs connus et offre ainsi une plus grande généralisation aux négatifs inconnus dans un scénario ouvert. Un classificateur CSA traditionnel est montré pour surpasser un classificateur de jeu ouvert dans un scénario largement fermé. Cependant, lorsque le degré d'ouverture augmente, la performance se dégrade rapidement, tandis que la performance de la nouvelle approche proposée pour l'CSA à cycle ouvert reste stable. Le classificateur SVDD apprend une hyper sphère uniquement à partir de données cibles. Tout en utilisant des données cibles uniquement pour la formation, ce classificateur est moins susceptible de se sur-ajuster aux données négatives connues et est donc plus fiable face à des données négatives inconnues. Une nouvelle approche basée sur une formulation de détection, un nouveau protocole et une métrique sont également introduits.

Une contribution supplémentaire concerne l'importance du réglage des paramètres du modèle. Deux méthodes sont comparées: l'une basée sur un critère basé sur la cible et l'autre, consciente des échantillons non ciblés. Selon le type d'applications CSA, un critère peut être préféré à l'autre: avec un niveau d'ouverture de 0,1, l'utilisation de l'ensemble de l'ensemble de la formation (cible et non-cible, si disponible) est bénéfique; lorsque l'incertitude est élevée, un critère basé uniquement sur les données d'entraînement semble plus robuste.

Les performances de l'algorithme SVDD sont corrélées au type de caractéristiques utilisées pour décrire chaque scène acoustique. A titre d'exemple, les fonctionnalités basées sur LBP montrent de meilleures performances pour certaines classes alors que les fonctionnalités basées sur MFCC conduisent à une meilleure fiabilité dans le cas de certaines autres classes. Plus généralement, les approches ouvertes peuvent être conçues avec n'importe quel classificateur, y compris les approches d'apprentissage en profondeur. Compte tenu des travaux récents qui montrent la vulnérabilité des architectures d'apprentissage en profondeur [156] à des échantillons spécifiquement conçus, une évaluation en circuit ouvert est nécessaire.

Les domaines comme classification d'image [157] et vérification de visage [158] ont commencé à remettre en question les évaluations en série fermée. Il existe des preuves que les approches actuelles d'apprentissage en profondeur montrent une performance trop optimiste et qu'elles ne sont pas robustes aux échantillons inconnus pendant les tests [141]. Dans le CSA, la variabilité inter et intra-classe est si élevée que le scénario de l'ouverture doit être pris en compte. L'évaluation future de l'ASC devrait tenir compte de ce scénario, car elle fournit un cadre d'évaluation plus proche de la réalité. Le classificateur SVDD est une solution possible au problème de l'ouverture, mais d'autres aspects de l'open-set devraient être étudiés dans de futures recherches:

- une meilleure caractérisation de chaque scène acoustique (par exemple en utilisant des architectures CNN pour extraire automatiquement des entités à partir de données);
- l'intégration du risque d'ouverture dans la minimisation des erreurs (par exemple, en remplaçant la fonction soft max par une fonction open max, adaptée à l'open-set [141]);
- l'exploitation d'échantillons non ciblés, lorsqu'ils sont disponibles (par exemple SVDD qui tire parti des échantillons non ciblés [159]);
- détection de nouveauté d'échantillons inconnus avec la définition automatique de nouvelles classes (par exemple détection de nouveauté basée sur les distances SVDD).

Étant donné que le scénario prédominant du scénario d'utilisation de la NCP est ouvert, il est à espérer que le point de vue proposé sur la NCP sera adopté par le milieu de la recherche à l'avenir.

Résultats en open-set

Les résultats sont illustrés pour les jeux de données DCASE 2013 et Rouen 2015 dans les Figs. 8 (a) et (b) respectivement. Les résultats du classificateur SVM sont illustrés par des profils en pointillés-bleu. Ceux du classificateur SVDD sont illustrés par des profils rouge. Pour chaque niveau d'ouverture, les résultats d'AUC sont moyennés sur toutes les classes avec le même niveau d'ouverture. Les barres verticales de la figure 7 reflètent l'écart-type AUC sur ces classes.

Des tendances similaires sont observées pour les deux ensembles de données. À mesure que l'ouverture augmente, la performance du classificateur SVM se dégrade, passant de 95% à 60% pour l'ensemble de données DCASE 2013 et de 90% à 50% pour l'ensemble de données Rouen 2015. En revanche, les résultats du classificateur SVDD restent relativement stables pour les deux ensembles de données, mesurant respectivement 80% et 85% de l'CSA pour les jeux de données DCASE 2013 et Rouen 2015.

Conformément aux résultats illustrés sur la figure 7, le classificateur SVDD surpasse le classificateur SVM. Cependant, un plus grand intérêt est la variation de performance pour différentes compositions de k classes négatives connues, encore illustrées en termes d'écart-type avec des barres verticales. Alors que la performance du classificateur SVM est affectée par une combinaison spécifique de k classes négatives connues, celle du classificateur SVDD est relativement non affectée.

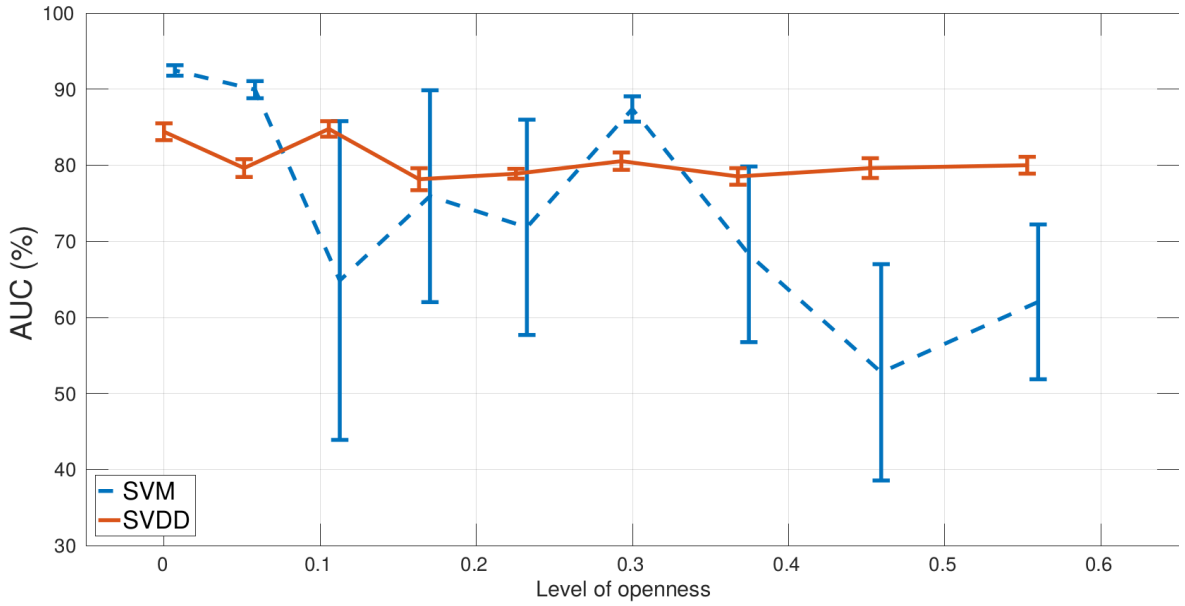
CONCLUSION

La recherche sur les particularités de l'CSA est le sujet que l'auteur a tenté d'étudier tout au long de la thèse. Des vues offertes à ce sujet, quelques conclusions générales sont dérivées. Ils sont détaillés comme suit:

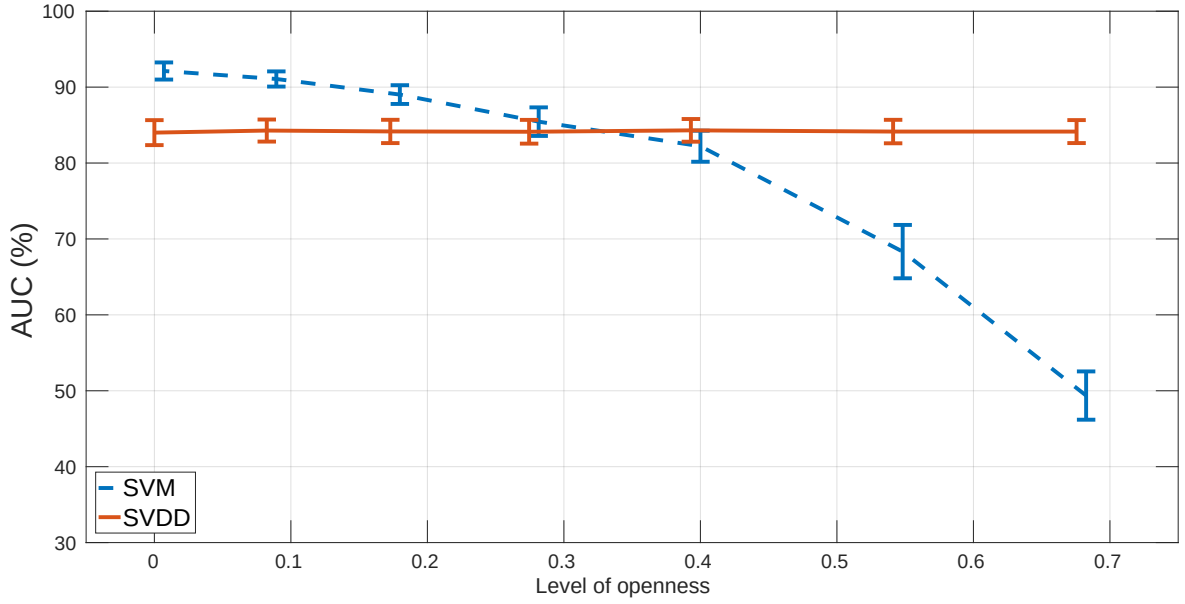
- CSA est une tâche très complexe d'un point de vue acoustique et taxonomique. Des scènes acoustiques similaires peuvent être classées sous deux concepts différents de haut niveau (par exemple, rue calme et parc) alors que le même concept peut contenir des scènes acoustiques très différentes (par exemple, une voiture sport et une voiture électrique) . Un ensemble de données complet qui capterait une variation serait coûteux et difficile à collecter. De plus, obtenir un accord de la communauté sur une taxonomie commune contient aussi un grand défi;

- une scène acoustique a une structure temporelle faible. Des sons proéminents peuvent apparaître dans n'importe quel ordre, de sorte que les méthodes qui modélisent une évolution temporelle ne seront pas adaptées pour représenter cette scène temporellement non structurée. Les systèmes basés sur les LBP ou les CNN reposent sur la présence de modèles spécifiques plutôt que sur leur évolution temporelle. Cette idée peut être vue sous la perspective « bottom-up » (Chapitre 2). Une perspective CSA regroupe différentes méthodes sous l'idée commune que les descripteurs audio de bas niveau (dans le cas des LBP et CNN, les patterns audio) composent toute la scène acoustique;

- une scène acoustique peut être caractérisée par des motifs spectro-temporels, qui extraient des informations de la représentation temps-fréquence. La nature de ces



(a)



(b)

Figure 7: Tracés de l'aire sous la courbe caractéristique de réception (AUC) par rapport à l'ouverture pour (a) ensemble d'évaluation DCASE 2013 et (b) ensembles de données Rouen 2015 pour les classificateurs SVM (profils en pointillés bleus) et SVDD (profils rouges-rouges). L'écart type est illustré par des barres verticales.

motifs peut être décidée a priori (par exemple, LBP) ou automatiquement extraite des données (par exemple CNN). Ce qui est différent des caractéristiques traditionnelles (par exemple MFCCs), est la corrélation significative en temps et en fréquence montrant qu'un descripteur unifié en temps et en fréquence peut obtenir un haut niveau de performance;

- CSA est un problème de classification ouvert. Avant d'effectuer une classification, un système CSA robuste doit d'abord déterminer si la scène est dans l'ensemble des classes connues ou l'identifier comme inconnue. Dans ce cas, le système CSA effectuera la détection avant la classification.

La recherche dans le domaine CSA repose toujours sur un scénario supervisé où les étiquettes et les données doivent être fournies. Ce paradigme supervisé est très inefficace lorsque la quantité de données d'apprentissage représente très mal la vraie variabilité des scènes acoustiques. Il est recommandé que les recherches futures étudient différentes approches, qui ne dépendent pas entièrement de données étiquetées. Dans ce sens, apprendre à partir de données faiblement annotées provenant d'autres domaines (vidéo, événement audio) est une option possible. Dans une perspective similaire, les approches d'apprentissage actif continu devraient être considérées comme une alternative aux approches supervisées existantes. Les solutions CSA peuvent avoir accès (via un microphone) à une quantité essentiellement infinie de données non étiquetées, mais l'étiquetage de ces données est coûteux. Une approche semi-supervisée telle que l'apprentissage actif [160] peut sélectionner un sous-ensemble de telles données à étiqueter automatiquement. Une fois que l'échantillon a été étiqueté (à partir d'une interaction avec l'utilisateur ou avec une autre source d'information), les modèles de scène peuvent être recyclés ou ajustés.

Une autre piste de recherche peut impliquer l'apprentissage par transfert [161, 162] entre domaines connexes. Par exemple, un système conçu pour détecter des événements audio pourrait être utilisé pour classer des scènes sans un réentraînement complet. Dans ce cas, l'objectif est de former un système CSA complet avec un jeu de données étiqueté à part entière utilisant les connaissances d'autres domaines (tels que la détection d'événements audio).

mpte tenu de l'application réussie de modèles temps-fréquence, d'autres recherches pourraient examiner un descripteur temps-fréquence-espace unifié incluant également les informations spatiales. Cela pourrait être fait en utilisant des entrées multicanaux pour les CNN. À ce jour, très peu de travaux ont abordé le problème CSA en utilisant des approches multi-microphones [54, 116] et, celui qui a un maximum de 2 microphones a considéré. Dans ce sens, les travaux futurs devraient tenir compte des multi-microphones ou des matrices de microphones.

En termes de recherche appliquée, on pense que les solutions à la tâche CSA nécessitent des avancées dans les domaines suivants:

1. ouvrir les protocoles et les métriques pour une évaluation publique future. Afin de réaliser le potentiel commercial de l'CSA et de réduire l'écart entre la recherche fondamentale et appliquée, la performance des solutions développées en laboratoire doit être confirmée par les utilisateurs ou par des tests sur le terrain;

2. robustesse aux enregistrements de haute qualité et de basse qualité. L'invariance d'une mauvaise qualité d'enregistrement doit faire l'objet d'une étude plus approfondie.

3. modéliser la complexité en termes de mémoire et de contraintes de calcul. Même en fournissant de bonnes performances de généralisation, les solutions d'apprentissage en profondeur peuvent contenir des millions de paramètres. Des travaux récents ont

été présentés pour réduire le nombre de paramètres dans un modèle CNN [163] et devraient être étudiés dans le contexte de l'CSA;

4. stratégies en temps réel impliquant l'analyse de l'audio en mode continu. En raison des ressources limitées, les périphériques de faible puissance ne peuvent pas stocker une quantité énorme d'audio. Cela signifie que les systèmes CSA en temps réel doivent utiliser un tampon d'échantillons audio limité pour extraire des fonctionnalités et effectuer des prédictions. Une approche candidate possible peut utiliser des topologies évolutives de réseaux neuronaux pour traiter des échantillons audio bruts directement [164] de manière continue en continu. Cela peut entraîner un compromis entre la flexibilité (en termes de nombre de paramètres réseau) et la performance;

5. dans le but de disposer d'un système fiable de reconnaissance du contexte acoustique, les signaux provenant de capteurs hétérogènes (caméra, capteurs de mouvement, capteurs de température) peuvent fournir une meilleure vue de l'environnement environnant. Dans cette hypothèse, il est clair que la fusion ou la combinaison de sources d'information hétérogènes pourrait constituer une future piste d'investigation.

Enfin, la communauté CSA s'est développée au cours des dernières années et attire maintenant l'intérêt des universités et de l'industrie. Afin de créer des applications CSA utiles et utilisables, une synergie entre la recherche fondamentale et appliquée doit devenir la voie standard pour la recherche future. On espère donc que l'analyse présentée dans cette thèse pourrait aider à orienter la recherche de l'CSA à l'avenir.

BIBLIOGRAPHY

---

[1] O. Räsänen, J. Leppänen, U. K. Laine, and J. P. Saarinen, "Comparison of classifiers in audio and acceleration based context classification in mobile phones," in *Proceedings of the 19th European Signal Processing Conference (EUSIPCO)*, pp. 946–950, Aug. 2011.

[2] R. Schafer, *The new soundscape: a handbook for the modern music teacher*. Berandol Music Limited Press, 1969.

[3] S. McAdams, *Recognition of sound sources and events*. Thinking in sound: the cognitive psychology of human audition, Oxford Univ. Press, May 1993.

[4] S. Handel, *Listening: An Introduction to the Perception of Auditory Events*. MIT Press, Jan. 1989.

[5] A. S. Bregman, *Auditory scene analysis: The perceptual organization of sound*. MIT press, Sept. 1994.

[6] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. Plumbley, "Detection and Classification of Acoustic Scenes and Events," *IEEE Transactions on Multimedia*, vol. 17, pp. 1733–1746, Oct. 2015.

[7] S. Graf, T. Herbig, M. Buck, and G. Schmidt, "Features for voice activity detection: a comparative analysis," *EURASIP Journal on Advances in Signal Processing*, vol. 2015, Dec. 2015.

[8] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, Nov. 2012.

[9] E. Cambria and B. White, "Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]," *IEEE Computational Intelligence Magazine*, vol. 9, pp. 48–57, May 2014.

[10] M. Schedl, E. Gómez, J. Urbano, and others, *Music information retrieval: Recent developments and applications*. Now Publ., Sept. 2014.

[11] P. Knees and M. Schedl, "Music retrieval and recommendation: a tutorial overview," in *Proceedings of the 38th ACM International Conference on Research and Development in Information Retrieval*, SIGIR '15, (New York, NY, USA), pp. 1133–1136, ACM, 2015.

[12] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, p. 1, 2013.

[13] S. Liang, X. Du, and P. Dong, "Public scene recognition using mobile phone sensors," in *Proceedings of the International Conference on Computing, Networking and Communications (ICNC)*, pp. 1–5, Feb. 2016.

[14] D. Battaglino, A. Mesaros, L. Lepauloux, L. Pilati, and N. Evans, "Acoustic context recognition for mobile devices using a reduced complexity SVM," in *Proceedings of the 23rd IEEE European Signal Processing Conference (EUSIPCO)*, Aug. 2015.

[15] G. T. Abreha, "An environmental audio-based context recognition system using smartphones," Master's thesis, University of Twente, August 2014.

[16] N. Kern and B. Schiele, "Context-aware notification for wearable computing," in *Proceedings of the 7th IEEE International Symposium on Wearable Computers (ISWC)*, pp. 223–230, 2003.

[17] S. Chu, S. Narayanan, C. c. J. Kuo, and M. J. Mataric, "Where am I? Scene Recognition for Mobile Robots using Audio Features," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 885–888, July 2006.

[18] H. G. Okuno, T. Ogata, and K. Komatani, "Robot Audition from the Viewpoint of Computational Auditory Scene Analysis," in *Proceedings of the International Conference on Informatics Education and Research for Knowledge-Circulating Society, (ICKS)*, Jan. 2008.

[19] T. Zhang and C. C. J. Kuo, "Audio content analysis for online audiovisual data segmentation and classification," *IEEE Transactions on Speech and Audio Processing*, vol. 9, pp. 441–457, May 2001.

[20] W. Zeng and M. Liu, "Hearing environment recognition in hearing aids," in *Proceedings of the 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 1556–1560, Aug. 2015.

[21] F. Beritelli, S. Casale, and S. Serrano, "Adaptive robust speech processing based on acoustic noise estimation and classification," in *Proceedings of the 5th IEEE International Symposium on Signal Processing and Information Technology*, pp. 773–777, Dec. 2005.

[22] C. Couvreur and P. Chapelle, "Utilization of expert knowledge in automatic classifiers of noise sources," vol. 18, pp. 2855–2858, 1996.

[23] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, pp. 1041–1044, ACM, 2014.

[24] E. Wang, H. Liu, G. Wang, Y. Ye, T.-Y. Wu, and C.-M. Chen, "Context recognition for adaptive hearing-aids," in *Proceedings of the 13th IEEE International Conference on Industrial Informatics (INDIN)*, pp. 1102–1107, July 2015.

[25] D. McMillan and A. Loriette, *Living with Listening Services: Privacy and Control in IoT*, pp. 100–109. Springer International Publishing, 2015.

[26] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Proceedings of the 24th IEEE European Signal Processing Conference, EUSIPCO*, pp. 1128–1132, 2016.

[27] B. Clarkson, N. Sawhney, and A. Pentland, "Auditory context awareness via wearable computing," *Energy*, vol. 400, no. 600, p. 20, 1998.

[28] R. Cai, L. Lu, and L.-H. Cai, "Unsupervised auditory scene categorization via key audio effects and information-theoretic co-clustering," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, pp. 1073–1076, Mar. 2005.

[29] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell, "Soundsense: Scalable sound sensing for people-centric applications on mobile phones," in *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, MobiSys '09, pp. 165–178, ACM, 2009.

[30] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley-Interscience, 2nd ed., 2000.

[31] P. S. R. Diniz, E. A. B. da Silva, and S. L. Netto, *Digital Signal Processing: System Analysis and Design (2nd Edition)*. Cambridge University Press, 2nd ed., July 2010.

[32] F. J. Valverde-Albacete and C. Peláez-Moreno, "100% Classification Accuracy Considered Harmful: The Normalized Information Transfer Factor Explains the Accuracy Paradox," *PLoS ONE*, vol. 9, Jan. 2014.

[33] A. Rakotomamonjy and G. Gasso, "Histogram of gradients of time-frequency representations for audio scene detection," *CoRR*, vol. abs/1508.04909, 2015.

[34] D. P. Ellis, "Prediction-driven computational auditory scene analysis for dense sound mixtures," in *Proceedings of the ESCA workshop on the Auditory Basis of Speech Perception*, 1996.

[35] K. El-Maleh, A. Samouelian, and P. Kabal, "Frame level noise classification in mobile environments," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, pp. 237–240, Mar. 1999.

[36] N. Sawhney and P. Maes, "Situational awareness from environmental sounds." Final Project Report for Modeling Adaptive Behavior (MAS 738), MIT Media Lab, 1997.

[37] V. T. Peltonen, A. J. Eronen, M. P. Parviainen, and A. P. Klapuri, "Recognition of everyday auditory scenes: potentials, latencies and cues," May 2001.

[38] V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, and T. Sorsa, "Computational auditory scene recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, pp. 1941–1945, 2002.

[39] A. Eronen, J. Tuomi, A. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, "Audio-based context awareness-acoustic modeling and perceptual evaluation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, pp. 529–532, IEEE, 2003.

[40] 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*, vol. 92, no. 6, pp. 865–874, 2006.

[41] J. Tardieu, P. Susini, F. Poisson, P. Lazareff, and S. McAdams, "Perceptual study of soundscapes in train stations," *Applied Acoustics*, vol. 69, pp. 1224–1239, Dec. 2008.

[42] A. J. Torija, D. P. Ruiz, and A. F. Ramos-Ridao, "Application of a methodology for categorizing and differentiating urban soundscapes using acoustical descriptors and semantic-differential attributes," *The Journal of the Acoustical Society of America*, vol. 134, pp. 791–802, July 2013.

[43] S. Chu, S. Narayanan, and C. C. J. Kuo, "Environmental sound recognition using MP-based features," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–4, Mar. 2008.

[44] Y. Lee, K. Kim, D. K. Han, and H. Ko, "Acoustic and visual signal based violence detection system for indoor security application," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE)*, pp. 737–738, Jan. 2012.

[45] I. Feki, A. B. Ammar, and A. M. Alimi, "Audio stream analysis for environmental sound classification," in *Proceedings of the International Conference on Multimedia Computing and Systems*, pp. 1–6, Apr. 2011.

[46] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, "Audio-based context recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 321–329, 2006.

[47] J. S. Downie, X. Hu, J. H. Lee, K. Choi, S. J. Cunningham, and Y. Hao, "Ten years of mirex (music information retrieval evaluation exchange): Reflections, challenges and opportunities," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 657–662, 2014.

[48] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "TIMIT acoustic phonetic continuous speech corpus," 1993.

[49] P. Mowlaee, R. Saeidi, and R. Martin, "Model-driven speech enhancement for multisource reverberant environment (signal separation evaluation campaign (sisec) 2011)," pp. 454–461, Springer, 2012.

[50] C. S. Greenberg, V. M. Stanford, A. F. Martin, M. Yadagiri, G. R. Doddington, J. J. Godfrey, and J. Hernandez-Cordero, "The 2012 NIST speaker recognition evaluation," in *Proceedings of the 14th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 1971–1975, 2013.

[51] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov, "ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Proceedings of the 16th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Sept. 2015.

[52] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, 1st ed., 2001.

[53] J. T. Geiger, B. Schuller, and G. Rigoll, "Recognising acoustic scenes with large-scale audio feature extraction and SVM," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[54] W. Nogueira, G. Roma, and P. Herrera, "Sound scene identification based on MFCC, binaural features and a support vector machine classifier," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[55] J. D. Krijnders and A. Gineke, "A tone-fit feature representation for scene classification," *Energy [dB]*, vol. 400, no. 450, p. 500, 2013.

[56] G. Roma, W. Nogueira, P. Herrera, and R. de Boronat, "Recurrence quantification analysis features for auditory scene classification," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[57] B. Elizalde, H. Lei, G. Friedland, and N. Peters, "An i-vector based approach for audio scene detection," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[58] D. Li, J. Tam, and D. Toub, "Auditory scene classification using machine learning techniques," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[59] M. Chum, A. Habshush, A. Rahman, and C. Sang, "IEEE aasp scene classification challenge using hidden Markov models and frame based classification," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[60] D. Li, I. K. Sethi, N. Dimitrova, and T. McGee, "Classification of general audio data for content-based retrieval," *Pattern Recognition Letters*, vol. 22, no. 5, pp. 533 – 544, 2001.

[61] K. Patil and M. Elhilali, "Multiresolution auditory representations for scene classification," *Cortex*, vol. 87, no. 1, pp. 516–527, 2002.

[62] J. Nam, Z. Hyung, and K. Lee, "Acoustic scene classification using sparse feature learning and selective max-pooling by event detection," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[63] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Audio context recognition using audio event histograms," in *Proceedings of 18th IEEE European Signal Processing Conference (EUSIPCO)*, pp. 1272–1276, 2010.

[64] A. Kumar, P. Dighe, R. Singh, S. Chaudhuri, and B. Raj, "Audio event detection from acoustic unit occurrence patterns.," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 489–492, 2012.

[65] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[66] E. Olivetti, "The wonders of the normalized compression dissimilarity representation," 2013.

[67] J. P. Zbilut and C. L. Webber, *Recurrence Quantification Analysis*. John Wiley Sons, 2006.

[68] J. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines," *Microsoft Research*, Apr. 1998.

[69] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[70] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 1995.

[71] D. P. W. Ellis, "PLP and RASTA (and MFCC, and inversion) in Matlab," 2005. online web resource.

[72] C. Chang and C. Lin, "libSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 1–27, 2011.

[73] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bulletin*, vol. 1, pp. 80–83, Dec. 1945.

[74] K. Kumar, C. Kim, and R. M. Stern, "Delta-spectral cepstral coefficients for robust speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4784–4787, 2011.

[75] E. Keogh and A. Mueen, "Curse of dimensionality," in *Encyclopedia of Machine Learning*, pp. 257–258, Springer, 2011.

[76] M. F. de Oliveira and H. Levkowitz, "From visual data exploration to visual data mining: a survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 3, pp. 378–394, 2003.

[77] L. van der Maaten and G. E. Hinton, "Visualizing high-dimensional data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[78] M. Wattenberg, F. Viégas, and I. Johnson, "How to use t-sne effectively," *Distill*, 2016. http://distill.pub/2016/misread-tsne.

[79] L. Van Der Maaten, "Accelerating t-SNE using tree-based algorithms.," *Journal of machine learning research*, vol. 15, no. 1, pp. 3221–3245, 2014.

[80] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[81] G. Roffo, "Feature Selection Library (MATLAB Toolbox)," *arXiv preprint arXiv:1607.01327*, 2016.

[82] Q. Gu, Z. Li, and J. Han, "Generalized fisher score for feature selection," in *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, UAI'11, pp. 266–273, AUAI Press, 2011.

[83] G. Xuan, X. Zhu, P. Chai, Z. Zhang, Y. Q. Shi, and D. Fu, "Feature Selection based on the Bhattacharyya Distance," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR)*, vol. 3, pp. 1232–1235, 2006.

[84] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[85] C. Reyes-Aldasoro and A. Bhalerao, "The Bhattacharyya space for feature selection and its application to texture segmentation," *Pattern Recognition*, vol. 39, pp. 812–826, May 2006.

[86] T. Kailath, "The divergence and bhattacharyya distance measures in signal selection," *IEEE Transactions on Communication Technology*, vol. 15, pp. 52–60, Feb. 1967.

[87] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transacrtions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[88] J. Burred, "Genetic motif discovery applied to audio analysis," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 361–364, March 2012.

[89] Y. Costa, L. Oliveira, A. Koerich, and F. Gouyon, "Music genre recognition using gabor filters and lpq texture descriptors," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications* (J. Ruiz-Shulcloper and G. Sanniti di Baja, eds.), vol. 8259 of *Lecture Notes in Computer Science*, pp. 67–74, Springer Berlin Heidelberg, 2013.

[90] G. B. Huang, H. Lee, and E. Learned-Miller, "Learning hierarchical representations for face verification with convolutional deep belief networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2518–2525, 2012.

[91] N. Chatlani and J. Soraghan, "Local binary patterns for 1-d signal processing," pp. 95–99, 2010.

[92] F. Alegre, A. Amehraye, and N. Evans, "A one-class classification approach to generalised speaker verification spoofing countermeasures using local binary patterns," in *Proceedings of the 6th International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp. 1–8, Sept 2013.

[93] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[94] T. Kobayashi and J. Ye, "Acoustic feature extraction by statistics based local binary pattern for environmental sound classification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3052–3056, May 2014.

[95] X. Yuan, J. Yu, Z. Qin, and T. Wan, "A sift-lbp image retrieval model based on bag of features," in *Proceedings of the IEEE International Conference on Image Processing*, 2011.

[96] J. Choi, H. Cho, J. Kwac, and L. S. Davis, "Toward sparse coding on cosine distance," in *Proceedings of the International Conference on Pattern Recognition*, 2014.

[97] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[98] D. Battaglino, L. Lepauloux, L. Pilati, and N. Evans, "Acoustic context recognition using local binary pattern codebooks," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct. 2015.

[99] Y. Petetin, C. Laroche, and A. Mayoue, "Deep neural networks for audio scene recognition," in *Proceedings of the 23rd IEEE European Signal Processing Conference (EUSIPCO)*, pp. 125–129, Aug. 2015.

[100] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in neural information processing systems*, pp. 1097–1105, 2012.

[101] T. N. Sainath, B. Kingsbury, A.-r. Mohamed, G. E. Dahl, G. Saon, H. Soltau, T. Beran, A. Y. Aravkin, and B. Ramabhadran, "Improvements to deep convolutional neural networks for LVCSR," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 315–320, 2013.

[102] L. M. Miller, M. A. Escabí, H. L. Read, and C. E. Schreiner, "Spectrotemporal receptive fields in the lemniscal auditory thalamus and cortex," *Journal of Neurophysiology*, vol. 87, no. 1, pp. 516–527, 2002.

[103] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proceeding of the European conference on computer vision*, pp. 818–833, Springer, 2014.

[104] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proceedings of the IEEE international conference on Acoustics, speech and signal processing (ICASSP)*, pp. 4277–4280, 2012.

[105] M. Li, T. Zhang, Y. Chen, and A. J. Smola, "Efficient mini-batch training for stochastic optimization," in *Proceedings of the 20th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, KDD '14, pp. 661–670, ACM, 2014.

[106] C. Huang, Y. Li, C. Change Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5375–5384, 2016.

[107] R. S. Sutton, "Two problems with backpropagation and other steepest-descent learning procedures for networks," in *Proceedings of the 8th conference on cognitive science society*, pp. 823–831, Erlbaum, 1986.

[108] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural networks*, vol. 12, no. 1, pp. 145–151, 1999.

[109] J. Schmidhuber, "Learning complex, extended sequences using the principle of history compression," *Learning*, vol. 4, no. 2, 2008.

[110] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research*, vol. 9, pp. 249–256, Jan. 2010.

[111] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.

[112] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv:1207.0580*, 2012.

[113] D. Battaglino, L. Lepauloux, and N. Evans, "Acoustic scene classification using convolutional neural networks," tech. rep., DCASE2016 Challenge, Sept. 2016.

[114] F. J. Huang and Y. LeCun, "Large-scale learning with svm and convolutional for generic object categorization," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 284–291, 2006.

[115] P. P. Brahma, D. Wu, and Y. She, "Why Deep Learning Works: A Manifold Disentanglement Perspective," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, pp. 1997–2008, Oct. 2016.

[116] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, "Submissions for DCASE-2016: a hybrid approach using binaural i-vectors and deep convolutional neural networks," tech. rep., DCASE2016 Challenge, Sept. 2016.

[117] V. Bisot, R. Serizel, S. Essid, and G. Richard, "Supervised nonnegative matrix factorization for acoustic scene classification," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2016.

[118] M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen, "DCASE 2016 acoustic scene classification using convolutional neural networks," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2016.

[119] J. Kim and K. Lee, "Empirical study on ensemble method of deep neural networks for acoustic scene classification," tech. rep., DCASE2016 Challenge, Sept. 2016.

[120] S. H. Bae, I. Choi, and N. S. Kim, "Acoustic scene classification using parallel combination of LSTM and CNN," tech. rep., DCASE2016 Challenge, 2016.

[121] H. Phan, L. Hertel, M. Maass, P. Koch, and A. Mertins, "CNN-LTE: a class of 1-x pooling convolutional neural networks on label tree embeddings for audio scene recognition," tech. rep., DCASE2016 Challenge, Sept. 2016.

[122] T. Lidy and A. Schindler, "CQT-based convolutional neural networks for audio scene classification and domestic audio tagging," tech. rep., DCASE2016 Challenge, Sept. 2016.

[123] A. Santoso, C.-Y. Wang, and J.-C. Wang, "Acoustic scene classification using network-in-network based convolutional neural network," tech. rep., DCASE2016 Challenge, Sept. 2016.

[124] L. Hertel, H. Phan, and A. Mertins, "Classifying variable-length audio files with all-convolutional networks and masked global pooling," tech. rep., DCASE2016 Challenge, Sept. 2016.

[125] Y. Xu, Q. Huang, W. Wang, and M. D. Plumbley, "Hierarchical learning for DNN-based acoustic scene classification," tech. rep., DCASE2016 Challenge, Sept. 2016.

[126] M. Mak and S. Kung, "Low-power SVM classifiers for sound event classification on mobile devices," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1985–1988, 2012.

[127] J.-C. Wang, J.-F. Wang, and Y.-S. Weng, "Chip design of mfcc extraction for speech recognition," *Integration, the VLSI Journal*, vol. 32, pp. 111–131, Nov. 2002.

[128] D. E. Knuth, *The Art of Computer Programming: Seminumerical Algorithms*, vol. 2. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 3rd ed., 1997.

[129] R. Koggalage and S. Halgamuge, "Reducing the number of training samples for fast support vector machine classification," *Neural Information Processing-Letters and Reviews*, vol. 2, no. 3, pp. 57–65, 2004.

[130] Y.-J. Lee and S.-Y. Huang, "Reduced Support Vector Machines: A statistical theory," *IEEE Transactions on Neural Networks*, vol. 18, pp. 1–13, Jan 2007.

[131] D. H. Mai and N. L. Chi, "Training data selection for Support Vector Machines model," *IPCSIT*, vol. 6, 2011.

[132] H. Cao, T. Naito, and Y. Ninomiya, "Approximate RBF Kernel SVM and Its Applications in Pedestrian Classification," in *Proceedings of the 1st International Workshop on Machine Learning for Vision-based Motion Analysis (MLVMA)*, Oct. 2008.

[133] S. Sigtia, A. M. Stark, S. Krstulović, and M. D. Plumbley, "Automatic environmental sound recognition: Performance versus computational cost," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2096–2107, 2016.

[134] T. Li, S. Zhu, and M. Ogihara, "Using discriminant analysis for multi-class classification: an experimental investigation," *Knowledge and information systems*, vol. 10, no. 4, pp. 453–472, 2006.

[135] D. Battaglino, A. Mesaros, L. Lepauloux, L. Pilati, and N. Evans, "Acoustic context recognition for mobile devices using a reduced complexity svm," in *Proceedings of the 23rd European Signal Processing Conference (EUSIPCO)*, pp. 534–538, Aug. 2015.

[136] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience, 9th ed., Mar. 1990.

[137] W. Sun, J. Qu, Y. Chen, Y. Di, and F. Gao, "Heuristic sample reduction method for support vector data description," *Turkish journal of electrical engineering & computer sciences*, vol. 24, pp. 298–312, 2016.

[138] W. J. Scheirer, A. Rocha, A. Sapkota, and T. E. Boult, "Towards open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 36, July 2013.

[139] A. Rabaoui, H. Kadri, Z. Lachiri, and N. Ellouze, "One-class svms challenges in audio detection and classification applications," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 1, pp. 1–14, 2008.

[140] L. P. Jain, W. J. Scheirer, and T. E. Boult, "Multi-class open set recognition using probability of inclusion," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Sept. 2014.

[141] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1563–1572, 2016.

[142] L. Lu, H.-J. Zhang, and H. Jiang, "Content analysis for audio classification and segmentation," *IEEE Transactions on Speech and Audio Processing*, vol. 10, pp. 504–516, Oct. 2002.

[143] A. Banerjee, P. Burlina, and C. Diehl, "A support vector method for anomaly detection in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 8, p. 2282, 2006.

[144] A. Ypma, D. Tax, and R. Duin, "Robust machine fault detection with independent component analysis and support vector data description," in *Proceedings of the 9th IEEE Signal Processing Society Workshop*, pp. 67–76, Aug. 1999.

[145] D. M. J. Tax and R. P. W. Duin, "Data domain description using support vectors," in *Proceedings of the European Symposium on Artificial Neural Networks*, pp. 251–256, 1999.

[146] B. Schiilkopf, "The kernel trick for distances," in *Proceedings of the 13th Conference on Advances in Neural Information Processing Systems*, vol. 13, p. 301, MIT Press, 2001.

[147] X. Wang, F.-l. Chung, and S. Wang, "Theoretical analysis for solution of support vector data description," *Neural Networks*, vol. 24, no. 4, pp. 360–369, 2011.

[148] L. Zhuang and H. Dai, "Parameter optimization of kernel-based one-class classifier on imbalance learning," *Journal of Computers*, vol. 1, no. 7, pp. 32–40, 2006.

[149] A. Theissler and I. Dear, "Autonomously determining the parameters for SVDD with RBF kernel from a one-class training set," in *Proceedings of the International Conference on Machine Intelligence (WASET)*, pp. 1135–1143, 2013.

[150] F. Li and H. Wechsler, "Open set face recognition using transduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1686–1697, 2005.

[151] D. P. W. Ellis, "PLP and RASTA (and MFCC, and inversion) in Matlab," 2005. Software available at `http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/`.

[152] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[153] I. B. Mohamad and D. Usman, "Standardization and its effects on k-means clustering algorithm," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 6, no. 17, pp. 3299–3303, 2013.

[154] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, pp. 427–437, July 2009.

[155] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," pp. 61–74, 1999.

[156] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 427–436, 2015.

[157] A. B. Bapst, J. Tran, M. W. Koch, M. M. Moya, and R. Swahn, "Open set recognition of aircraft in aerial imagery using synthetic template models," in *Proceedings of the SPIE Security + Defence conference*, vol. 10202, pp. 1020206–1020206–18, 2017.

[158] M. Günther, S. Cruz, E. M. Rudd, and T. E. Boult, "Toward open-set face recognition," *CoRR*, vol. abs/1705.01567, 2017.

[159] Z. Wang, Z.-S. Zhao, and C. Zhang, *SVM-SVDD: A New Method to Solve Data Description Problem with Negative Examples*, pp. 283–290. Berlin, Heidelberg: Springer, 2013.