From the Institute for Signal Processing
of the University of Lübeck
Director: Prof. Dr.-Ing. Alfred Mertins

Audio Event Detection, Classification, and Beyond

Dissertation
for Fulfillment of
Requirements
for the Doctoral Degree
of the University of Lübeck

from the Department of Computer Sciences

Submitted by

Quoc Huy Phan
from Ben Tre, Vietnam

Lübeck, 2017

First referee:	Prof. Dr.-Ing. Alfred Mertins
Second referee:	Prof. Dr. rer. nat. Heinz Handels
Chairman:	Prof. Dr. rer. nat. Ralf Möller
Date of oral examination:	August 23rd, 2017

Approved for printing. Lübeck, May 4th, 2018

Zusammenfassung

Diese Arbeit befasst sich mit neuen Verfahren für die akustische Ereigniserkennung in kontinuierlichen Datenströmen sowie der Klassifikation von isolierten Audioereignissen. Die vorgeschlagene Erkennungsmethode basiert auf einem Regressionsansatz, bei dem die einzelnen Ereignisse in Segmente zerlegt werden, deren Positionen relativ zum Anfang und Ende des Ereignisses mit Hilfe eines Regressionsmodells geschätzt werden. Auf der Basis des gelernten Modells kann dann für ein unbekanntes Segment eine Schätzung des Anfangs- und Endzeitpunktes des Ereignisses durchgeführt werden. Weiterhin wird eine *Pipeline*-Architektur mit Unterstützung mehrerer Ereignisklassen eingeführt. Die experimentellen Ergebnisse auf dem ITC-Irst Datensatz zeigen, dass der neue Ansatz die bislang veröffentlichten Methoden in Bezug auf die Erkennungsleistung deutlich übertrifft. Zudem wird die sequenzielle Natur der Audiosignale genutzt und eine Detektion bei nur partieller Beobachtung (vor Beendigung des Ereignisses) ohne Verschlechterung der Erkennungsleistung ermöglicht. Ebenso erlaubt die Methode eine Zusammenführung partieller Informationen aus unterschiedlichen Kanälen (*multi channel fusion framework*), die zu einer weiteren Verbesserung der Erkennungsleistung führt.

Für das Klassifikationsproblem werden drei lernbare Repräsentationen eingeführt: Audiosegmente (*audio phrases*), Regressionsverfahren (*bank-of-regressors*) und generische sprachbasierte Repräsentationen. Ziel der Audiosegmente und Regressionsverfahren ist eine Codierung der temporalen Struktur der Ereignisse, mit der die Einschränkungen der bekannten *Bag-of-words*-Modelle umgangen werden sollen. Die Audiosegmente enthalten mehrere Lauteinheiten, die in einer gemeinsamen Struktur zusammengefasst sind. Um das Problem der hohen Dimensionalität der Audiosegmente zu umgehen, wird eine Methode zum Lernen diskriminativer kompakter Wörterbücher eingeführt. Das Gesamtmodell besteht aus einer Aneinanderreihung der klassenweisen Regressionsmodelle. Die Antworten der Modelle bilden Merkmalsvektoren, mit denen die Ereignisse charakterisiert werden. Jeder Eintrag eines gelernten Merkmalsvektors gibt dabei an, wie gut das

Ereignis mit der temporalen Struktur des zugehörigen Regressormodells übereinstimmt. Die Ensembles von Regressionsanalysen erlauben die Codierung von gemeinsamen Merkmalen zwischen unterschiedlichen Zielklassen. Experimente auf vier unterschiedlichen Datensätzen zeigen, dass die Repräsentationen mittels Audiosegmenten und Regressionsmodellen eine Erkennungsleistung liefern, die über dem bisherigen Stand der Technik liegt. In der generischen sprachbasierten Repräsentation werden Sprachmuster, wie zum Beispiel Ensembles von Phonen, als akustische Basisstruktur benutzt. Ein Audioereignis wird durch seine Ähnlichkeit zu diesen Mustern beschrieben. Diese Ähnlichkeiten können mit Hilfe von Multiklassen-Sprachklassifikatoren oder hierarchischen Bäumen von binären Klassifikatoren gewonnen werden. Die Repräsentationen zeigen sehr gute Klassifikationsgenauigkeiten in den durchgeführten Experimenten. Im Gegensatz zu anderen lernbaren Merkmalen sind sie generisch. Das bedeutet, dass ein Merkmalsvektor für verschiedene Datensätze ohne ein wiederholtes Training benutzt werden kann.

Aufbauend auf einer genauen Untersuchung der Unterschiede in den Ergebnissen der Ereigniserkennung und –klassifikation wird eine neue Methode für die Reduktion der Falsch-Positiv-Rate vorgeschlagen. Im modifizierten Verfahren wird ein leistungsstarker Klassifikator nachgeschaltet, mit dem falsch-positive Entscheidungen erkannt und zurückgewiesen werden können. Eine empirische Studie mit verschiedenen Kombinationen von Erkennern und Klassifikatoren zeigt eine konsistente Leistungsverbesserung in Bezug auf den F1-Score.

Abstract

The aim of this work is to contribute to the development of both audio event detection in continuous streams and isolated audio event classification. The improvement in the quality of audio event detection is achieved using a regression-based approach. The idea is to model relative positions of the audio segments, into which event instances are decomposed, to the event onsets and offsets using a regression model. Via the learned regression model, an unseen audio segment will be used to make estimations of where the onset and offset of the target event will likely be in a test audio signal. A detection pipeline supporting multiple categories at the same time is also introduced. The experimental results on the ITC-Irst dataset demonstrate superiority of the proposed approach over the common ones used in literature. It is further shown that the proposed approach accommodates the sequential nature of audio streams easily, allowing detection of audio events even when only partial durations are observed (i.e. early detection) without losing any overall accuracy. Finally, the proposed approach also offers a simple and efficient multi-channel fusion framework to leverage the partial information of distributed microphones to improve the detection performance.

To tackle the isolated audio event classification problem, three different learned representations are introduced: audio phrases, bank-of-regressors, and speech-based generic representations. The audio phrases and bank-of-regressors aim at encoding temporal structure of audio events. The concept of audio phrase is proposed in order to overcome the limitations of the well-known bag-of-words model. To accomplish this, an audio phrase combines multiple audio words to capture their dependency. Moreover, a method to learn a discriminative compact codebook is further introduced to remedy the high dimensionality of high-order audio phrases. Concerning the bank-of-regressors representation, the class-wise regression models previously used in the detection task are stacked into a bank and their responses to an audio event are used to characterize the event. Intuitively, each entry of a learned feature vector can be interpreted as how the input event aligns with the temporal structure modeled by the corresponding regressor. Stacking multiple

regressors allows shared features between different target event classes to be encoded. Experiments on four different datasets show state-of-the-art performance obtained by the audio phrases and bank-of-regressors representations. Regarding the speech-based generic representations, employing a set of speech patterns (i.e. phone triplets) as basic acoustic concepts, an audio event instance is represented by its similarities to these speech patterns. The speech similarities are obtained via either a simple multi-class speech classifier or a label-tree based hierarchy of binary classifiers. These representations show good classification accuracies on the experimental datasets. Additionally, opposing to other learned features, they are generic. That is, the feature extractor can be used for different audio event datasets without re-training.

Going beyond the audio event detection and classification tasks, an analysis of their dissimilarities is conducted. This analysis is then leveraged for a generic improved detection pipeline which supports false positive reduction for the detection task. In this enhanced pipeline, a certain detection system is augmented with a verification step where a high-quality classifier is employed to verify and reject detected false positives. An empirical study on various combinations of detectors and classifiers demonstrates consistent improvements on overall detection performance in terms of F1-score.

Acknowledgments

Firstly, I would like to sincerely thank my supervisors, Prof. Alfred Mertins and Prof. Jens-Martin Träder, and my advisor, Prof. Erhardt Barth for their invaluable advice and guidance in my research. I am deeply grateful to Prof. Mertins for the time that he has spent discussing ideas and revising papers as well as for his kindly care and help in my personal life.

Second, I would like to thank my colleagues at the Institute for Signal Processing, University of Lübeck. My research has been benefited a lot from discussing a wide range of research topics. I am also thankful to them for making the Institute a friendly and comfortable working environment.

Finally, this work could not have been done without the constant love and emotional support from my wife and our sons. I am blessed to have them brightening up every single day of my life.

For my family: Tiên, Dương, and Minh

Contents

1 Introduction

1.1 Motivation

Real-life acoustic environments involve different types of sounds: speech, music, and others. In the general field of "machine hearing" [135], it is unquestionable that understanding human speech is the most important challenge in building a system which can perform a variety of hearing tasks as good as human ears. However, such a goal would be unachievable as long as that system is unable to understand nonspeech sounds as humans do. These sounds cover a wide range of audio concepts which can be roughly categorized as in Table 1.1.

Computational analysis of nonspeech sounds is becoming a more and more active research area [17, 135, 231] which can be sub-divided into smaller areas such as audio scene recognition [11, 143, 208], audio event detection/classification [140, 143, 180, 208, 212], urban sound classification [7, 177, 192], animal sound analysis [20, 101, 207, 237] to mention a few. Audio event detection/classification recently gained great attention due to their potential applicability. Several international challenges have been organized, aiming at standardizing datasets and evaluation metrics for these tasks. They includes CLEAR 2006 [214], CLEAR 2007 [212], DCASE 2013 [74, 208], and DCASE 2016 [1, 143]. There is a significant increase in the number of participants of these campaigns over recent years.

Audio events can be thought of as acoustic "objects" produced by physical events taking place in the surrounding environments. They are also referred to as audio concepts [59, 184], sound events [48, 140, 221], and acoustic events [144, 244] in literature. Audio events are defined as concrete and temporally bounded units of sounds, such as phone ringing, footsteps, laughing, or coughing, etc. Because audio events may carry rich information about physical events such as actions and objects, the ability to understand the implicit information is important for many applications.

Security surveillance: The goal of an automatic surveillance system is to recognize potentially dangerous situations and keep security personnel or authorities

Table 1.1: Categorization of nonspeech and nonmusic audio sounds.

Sound Type	**Examples**
Human nonspeech sounds [44, 56, 194, 208, 214]	coughing, snoring, laughing, etc.
Animal sounds [20, 21, 101, 207, 237]	bird sounds, frog calls, etc.
Ambient sounds/soundscapes [63, 87, 177]	rain, sea waves, etc.
Interior/domestic sounds [180, 205, 208, 214]	door knock, glass breaking, chair moving, etc.
Exterior/urban sounds [153, 177, 192]	siren, car horn, etc.

informed about them. Although video is the most popular tool for surveillance, it likely fails to capture audio events such as glass shattering, screaming or gunshots. Acoustic information will help tremendously in these scenarios. Therefore, an audio-based event detection system has not only been used as a complement to video-based ones [6, 226, 239], but also as a standalone surveillance system [28, 40, 65, 121, 183].

Ambient assisted living: In this domain, an acoustic monitoring system is applied to anticipate the needs of inhabitants unobtrusively while maintaining their safety and comfort [30, 81, 225]. Acoustic sensors can capture indoor sounds and retrieve useful information which can enhance the quality of daily living [201, 224, 225] or compensate one's disabilities through home automation [15, 141]. In addition, distressed situations (e.g. falls, screams, or intense coughs) can also be automatically detected and appropriate actions can be subsequently triggered [80, 81, 197]. They are particularly useful in the context of in-home care to the elderly. Moreover, acoustic sensors are small and unobtrusive. Therefore, they compromise privacy infringement better than visual ones [70, 142].

Video/multimedia analysis: Audio event detection and classification can serve as an important complement to video event detection [5, 19, 58, 210, 234]. There exist various situations, such as crowd cheering or fire alarm, that visual cues alone struggle to identify. However, these events can be detected easily with audio cues. Integration of both data sources has resulted in improved accuracy for video event detection [34, 110, 132]. In addition, according to [135], one can have machines listening to videos and learning from them to "categorize, organize, and

index" them better. Promising results have been reported for video classification [19, 96, 128], searching [10, 111, 160, 233], and summarization [62, 109].

Ecosystem monitoring: There is increasing interest in using audio recordings (i.e. bioacoustics) to monitor animal population densities and migration patterns [16, 138]. They can also be used to monitor overall ecosystem health as well as to help people understanding animal calls [16]. For these purposes, the audio modality has been found to be well-suited. For example, many birds are much more clearly detectable by sound than video and other indicators [207]. Many audio event detection/classification systems have been developed for the analysis of bird songs [20, 21, 206, 207], frog calls [101, 236, 237], and many other animal sounds [86, 126, 146].

Other applications: Audio event detection/classification have also been employed to improve the naturalness of the interaction between humans and machines [91, 166] or to enhance voice activity detection [36]. In addition, in the cousin task of acoustic scene recognition [1, 11, 208] (i.e. determining whether the surrounding environment is an office, street or train station using audio signals), audio events produced by a scene can be used as its signature. Therefore, audio event detection systems have also been utilized to provide evidence for the scene recognition task [11, 92].

Despite of their great potential for many applications, compared to the mature field of automatic speech recognition (ASR), audio event detection/classification are still in their infancy. The classification task [49, 51, 97, 140, 214, 221], whose goal is to assign a label to a segmented event instance, appears to be easy at a glance. However, it is actually not the case for two reasons. Firstly, many audio events exhibit strong temporal structures that require explicit modeling. For example, a "car passing by" event can be heard fading in from one ear and then fading out on another ear after reaching the peak. Unfortunately, their temporal structures are different from those of speech in the manner that they are more complex with wider variety in frequency content and duration. It turns out that these structures can not be simply modeled using techniques adapted from speech modeling. The evidence is in their unsatisfactory performance on the audio event classification task [214]. Secondly, representation learning with deep neural networks have tremendously excelled in many speech and music related problems, such as acoustic modeling [4, 46, 89, 98, 191], speech synthesis [150, 250], and music classification and recommendation [38, 106, 249]. However, they have

just recently emerged on this task [5, 26, 163]. Furthermore, their effectiveness is unconvincingly justified due to the small sizes of their employed datasets.

The task of audio event detection is even more challenging than the classification one. Its goal is to determine both the identities and the time intervals of the event instances occurring in continuous signals [26, 143, 208, 214]. In this task, one needs to not only distinguish between different event categories of interest but also to tell them apart from the noisy background which usually consists of various kinds of sounds. In addition, in detection, the temporal boundaries of the event instances are unknown in advance. Therefore, one does not have access to the global context of event instances as in classification but needs to rely on unreliable local audio features to make inference. Two popular approaches have been proposed for the audio event detection task: detection-by-classification [85, 120, 180, 212, 214] and the ASR-based framework [93, 107, 144, 208, 242, 244]. The former employs classification models trained on isolated events to classify continuous signals in a sliding window fashion for detection. While these classification models are limited in capturing the temporal configurations of audio events, it is also hard to determine a good window size to properly handle the high intra- and extra-class variation of audio event durations. Moreover, even though the latter has been shown working well on speech, it is inefficient to capture the temporal development of audio event signals which is much more complex than that of speech [48].

On the other hand, most (if not all) existing works have been trying to improve the overall performance of a classification/detection system towards an oracle one. There are various important issues that have not been considered or explicitly addressed. Firstly, for the classification task, representations for audio events have been derived based on the analysis of the target signals per se. However, there is still lack of a general way to represent the audio signals and specifically a universal descriptor for them. Such a generic representation would be very helpful for solving various audio analysis tasks in a unique way, regardless of data sources. Secondly, regarding the detection task, since a temporal audio event has duration, a question arises whether it can be detected early even when only a partial duration of it has been observed by the system. The ability of early detection will greatly improve the quality of services, especially in security and safety related applications [81, 156, 225] as well as in human-robot interaction [91, 99, 166]. Thirdly, none of previous works have tackled the issue of false positive reduction explicitly. Since a detection system usually relies on unreliable local features, occurrences of false

positives are very likely. These false alarms can be reduced to enhance the overall performance of a given detection system with less effort given its state-of-the-art performance.

1.2 Contributions

This thesis proposes novel solutions to model temporal structures of audio events. These approaches are employed to tackle both detection and classification tasks. Apart from that, the above-mentioned relevant issues will also be addressed. This work consists of six main contributions (three on audio event detection and three on audio event classification):

Random regression forests based audio event detection system. Given an event instance decomposed into a sequence of small segments, the segments should be on a particular arrangement due to the temporal structure of the event. Therefore, a certain segment can be used to estimate the positions of the event onset and offset. This motivates the idea of training a model to encode the relative positions of the audio segments with respect to the event onsets and offsets of the training examples. The model is formulated as a regression problem and resolved with random regression forests [42]. During testing, via the model, test audio segments will be used to make estimations of where in time the event onsets and offsets are. As soon as the onset and offset of an event are determined, it is jointly detected and segmented from the continuous signal. This contribution has been published in the IEEE/ACM Transactions on Audio, Speech and Language Processing journal [173], and in the Interspeech 2014 conference [175].

Early detection ability. A temporal audio event has a duration. A detection system that has early detection ability is able to detect the event as soon as possible even when a partial of it is observed. Yet, enabling this capability should not result in deterioration of the overall performance of the system. Simultaneous achievement of these two strict goals requires the monotonicity property of the detection function [99]. While this elegant property cannot be assured by a naive solution that simply detects a partial event, the proposed regression forests based audio event detection system is mathematically proven to fulfill this requirement. This contribution has been published in the ICME 2015 conference [170].

Additive multi-channel fusion framework. A simple yet efficient fusion framework is proposed to leverage spatial information of distributed microphones

to improve accuracy for audio event detection. A regression forest detector as mentioned above is developed for each microphone. Afterwards, the fusion system additively assembles confidence scores of the channel-wise detectors to gain evidence about occurrence of a target event. The work on the multi-channel fusion framework has been published in the WASPAA 2015 workshop [168].

Generic detection pipeline for false positive reduction. Based on the presented analysis of dissimilarities between the classification and detection tasks, an improved generic detection pipeline is derived. This pipeline enhances a typical one by appending a verification step in which a high-quality event classifier is employed to verify unreliable outputs of the detection system and reject false positives. Although consistent improvements are empirically shown with numerous combinations of event detectors and event classifiers, using the proposed detection system based on regression forests is more convenient and advantageous. Apart from its state-of-the-art performance, one can derive two high-level representations for training audio event classifiers (i.e. audio phrases and bank-of-regressors mentioned below) using the existing components of the detection system. While the classifiers trained on these in-situ representations offer state-of-the-art results on the classification task, using them as a verifier avoids the cost of learning an additional one.

Bag-of-phrases representation. Bag-of-words models which are widely used for audio event classification consider independent audio words and are, therefore, unable to take the structural information into account. The concept of audio phrases, which are defined as sequences of multiple audio words, are introduced to overcome this. The bag-of-phrases representation is able to capture the relationship between the isolated audio words and thus encodes a certain degree of structural information. However, the high dimensionality of this representation, which grows exponentially with the order of phrases, hinders its practicality at high orders. A compact codebook learning procedure using a discriminatively trained classifier is further proposed to cope with this issue. This contribution has been published in two conferences: EUSIPCO 2014 [167] and EUSIPCO 2015 [169].

Bank-of-regressors representation. As mentioned previously for the detection task, a random regression forest can model the temporal structure of event instances of a target category. Its response on an unseen event instance, therefore, quantifies how the event aligns to the temporal configuration of the category. A representation learning scheme is then proposed by stacking the class-wise re-

gressors on a bank and using their responses on an event instance as structural features to represent the event. The learned representation is able to encode shared features between different event categories of interest while being compact and highly discriminative. The work on the bank-of-regressors representation has been published in the ICASSP 2016 conference [171]

Generic representations based on speech similarity. Considering speech patterns (e.g. speech words or phone triplets) from an external speech database as basic acoustic concepts, a generic representation is proposed for audio events by measuring their similarities to speech patterns. That is, the event instances are embedded into the space of the speech similarities for representation. These similarities are obtained as the classification probabilities when evaluating the trained speech pattern classifier on the target event instances. Two classification schemes are studied for this purpose: the flat one with a single multi-class classifier and the hierarchical one with an automatically learned label tree. This representation is generic since once the speech classifiers have been learned, they can be used to extract features for any audio events from different data sources without re-training. This contribution has been published in the Interspeech 2015 conference [174] and in the IEEE/ACM Transactions on Audio, Speech and Language Processing journal [172].

1.3 Structure of the Thesis

The remaining chapters of this thesis are summarized as follows.

Chapter 2 presents a comprehensive literature review on audio event detection and classification. Different perspectives and open issues on these tasks are also discussed in this chapter.

Chapter 3 is dedicated to a novel detection system. The detection system based on random regression forests is proposed to estimate target event onsets and offsets. Experiments are also elaborated to analyze its performance as well as demonstrate its superiority over the baseline detection systems based on common approaches.

Chapter 4 proves the monotonicity property of the detection function of the proposed detection system in Chapter 3. After that, the additive multi-channel fusion framework is proposed. The extended experiments are also described to demonstrate the system's efficiency.

Chapter 5 focuses on the classification task. Two learned representations for audio events including the audio phrases and the bank of regressors are elaborated. State-of-the-art performances obtained by these representations are demonstrated on four different datasets.

Chapter 6 presents generic representations for audio events using the similarities between the audio events and basic speech patterns. Similarities derived from a "flat" multi-class speech classifier and a hierarchy of binary speech classifiers are both investigated. While these generic representations alone show good classification performance, they can also act as valuable external features to improve an existing system.

Chapter 7 revisits the detection tasks to deal with false positive reduction. The improved detection pipeline with an appended verification step is described in this chapter. Various classifiers (i.e. those proposed in this thesis and other baseline classifiers), are employed as a verifier. Coupling them with different event detectors (i.e. the proposed detector in Chapter 2 and the baseline detectors) shows consistent improvements in detection performance.

The summary of this thesis and possible future work are discussed in the final chapter.

2 Literature Review

Occurring physical events, such as a chair moving or a crowd cheering, produce signature sounds, which carry rich information about them. These sound events are concrete units of sound and temporally bounded to the beginning and ending time of the physical events. Furthermore, the sound can be captured using acoustic sensors. Signals recorded by the sound sensors can be analyzed to extract sound events which in turn help to gain knowledge about the physical events. Previous works on audio event detection and classification can be characterized with respect to different perspectives. This chapter is to investigate audio event representations, detection/classification algorithms, temporal structural encoding approaches, overlapping event handling, and fusion of multiple data sources. Finally, the open issues are addressed to motivate the development of this thesis.

2.1 Representations

The purpose of the feature extraction stage is to transform a redundant audio waveform into a compact representation prior to the stage of detection and classification. In general, audio events can be represented using any audio features which are used to describe an audio signal (e.g. those in [131, 165]). However, a good representation needs to minimize the distance between event instances of the same class, while maximizing the distance between those of different classes.

2.1.1 Segment-Wise Features

For detection and classification algorithms that rely on segment-wise processing, a fixed feature vector needs to be derived for each audio segment. Low-level hand-crafted features are usually employed for this purpose. The most commonly used features are those borrowed from speech recognition, for example Mel-frequency cepstral coefficients (MFCCs) [88, 93, 143, 144, 158, 230], log-frequency filter bank coefficients [22, 212, 214], and perceptual linear prediction (PLP) coefficients [181,

222]. Besides that, various frequency-domain features can be also enumerated, such as short-time Fourier transform (STFT) magnitude, spectral centroid, spectral roll-off, spectral shape statistics, spectral slope, spectral flatness, spectral flux, and spectral correlation [3, 139, 216] to mention a few. These cepstral and frequency features are capable to capture the perceptual and harmonic information of the audio signal. Usually, they are used alongside time-domain features, for example short-time energy, zero crossing rate, etc [139, 216]. Features based on wavelet transforms [190, 228], Gammatone filterbanks [48, 227], Gabor filterbanks [195, 196], and stabilized auditory images [136, 140] have also been investigated. A more detailed review on these features can be found in [3].

Apart from that, segment-wise features can be learned as well. If one considers the features learned at different layers of a deep network [26, 97, 140, 178], they richly range from mid- to high-level features. They are resulted when each layer of the network accumulates information from the layer beneath to form more complex features. Particularly, Hertel et al. [97] demonstrated that raw waveform can also be used as inputs to train such a deep network although the obtained results are inferior to those trained on magnitude spectral features. However, this effect is likely due to insufficient training data given the small sizes of the employed datasets in [97].

2.1.2 Event-Wise Features

Event-wise features are used when one needs to represent a whole event instance as a feature vector. They are not only particularly important for the classification task but also for the detection task when a detection system relies on event-wise classification models (i.e. detection by classification [85, 180]). Intuitively, a low-level feature vector representing an event instance can be computed using statistics, e.g. mean and standard deviation, over its constituent segment-wise feature vectors [48, 139, 212, 214]. However, with the rapid advance of machine learning techniques, the automatic feature learning approach is becoming more and more common. Importantly, the mid-level and high-level features learned on top of low-level features usually enjoy better discrimination power than that of the low-level ones.

Bag-of-words (BoW) models have been widely used for audio event representation [7, 28, 66, 117, 119, 160, 193]. In this method, the training event instances are first decomposed into multiple segments and a codebook is then learned using

segment-wise feature vectors, for example using k-means clustering [28, 66, 160] or a Gaussian Mixture Model (GMM) [117, 119, 180]. Each code word of the codebook is represented by the centroid of a cluster (in case of k-means) or a Gaussian component (in case of GMM). A segment-wise feature vector is then matched to a code word in the learned codebook with a certain weight. The matching weight can be "hard" (i.e. with k-means) or "soft" (i.e. with GMM). The descriptor for a whole audio event is finally produced by accumulating the matching weights into a vector, which has its size equal to the codebook size.

Due to the fact that the BoW models cannot capture the dependency of individual audio segments, several extensions have been further proposed to account for this shortcoming. Plinge et al. [180] considered a pyramid BoW model. This model aims at encoding the temporal layout at each pyramid level. A sequence of audio segments is firstly split into hierarchical cells for each of which a BoW representation is then computed. The final representation is formed by concatenating the BoW representations of all cells. Motivated by the n-gram models in language modeling [209], BoW models have also been extended by integrating the context of multiple consecutive audio segments [85, 161]. Other codebook-based representations have also been attempted, such as sparse coding [103, 134], non-negative matrix factorization (NMF) [41, 74], and exemplar-based coding [71, 133].

2.2 Audio Event Detection (AED)

The goal of an AED system is to determine two things about a target event: (1) where the event happens in a continuous audio signal and (2) its identity. The expected outputs of an AED system are illustrated in Figure 2.1. From the algorithmic viewpoint, two dominant trends have been seen in previous works. The first one relies on the conventional ASR framework [93, 107, 144, 156, 196, 208, 242, 244] and the second one is based on a detection-by-classification scheme [85, 120, 158, 180, 195, 214, 215].

2.2.1 ASR Framework Based Approach

Since the audio event detection task is analogous to the continuous speech recognition one, the ASR framework [154, 182] has been adapted for the event detection task [93, 107, 144, 156, 196, 208, 242, 244]. Under this framework, the audio events are treated like words in speech. This approach has been prevalently used

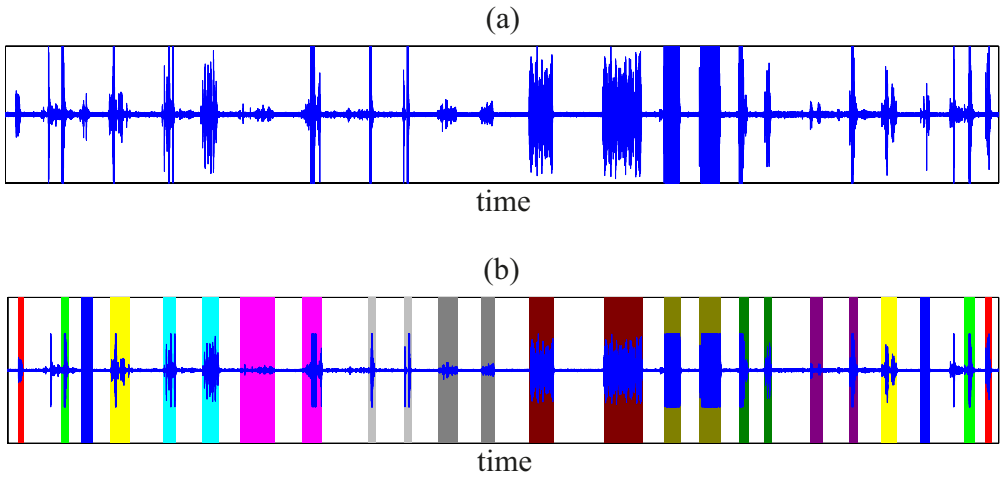

Figure 2.1: Detection of audio events from an audio signal: (a) the input audio signal and (b) the oracle AED output. The colors represent different event categories of interest.

in previous challenges, especially CLEAR 2006 [214], CLEAR 2007 [212], and DCASE 2013 [74, 208]. Typically, the detection systems following this approach can be divided into three stages. First, frame-level features, such as MFCCs, are extracted and their states are then modeled by GMMs. Second, Hidden Markov Models (HMMs) are employed to model the state sequences. Finally, during testing, the target event is detected and segmented by finding the state sequence with a maximum likelihood given the frame-wise feature vector sequence of the test audio signal.

Formally, following the ASR framework [182], the AED task can be formulated as:

$$\begin{aligned} \hat{\mathbf{E}} &= \arg\max_{\mathbf{E}} P\left(\mathbf{E} \,|\, \mathbf{X}\right) \\ &= \arg\max_{\mathbf{E}} P\left(\mathbf{X} \,|\, \mathbf{E}\right) P(\mathbf{E}). \end{aligned} \tag{2.1}$$

That is, given the acoustic observation or frame-wise feature vector sequence $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ of size N, the goal is to find the corresponding sequence $\hat{\mathbf{E}} = (e_1, e_2, \ldots, e_M)$ of M events that maximizes the posterior probability $P(\mathbf{E} \,|\, \mathbf{X})$. $P(\mathbf{E})$ denotes the prior probability of the event sequence $\mathbf{E}$, which is usually assumed to be identical for all event sequences. Bigram models, which account for the probability of a certain event conditioned on the previous event in the event sequence, were also found useful for the AED task [242].

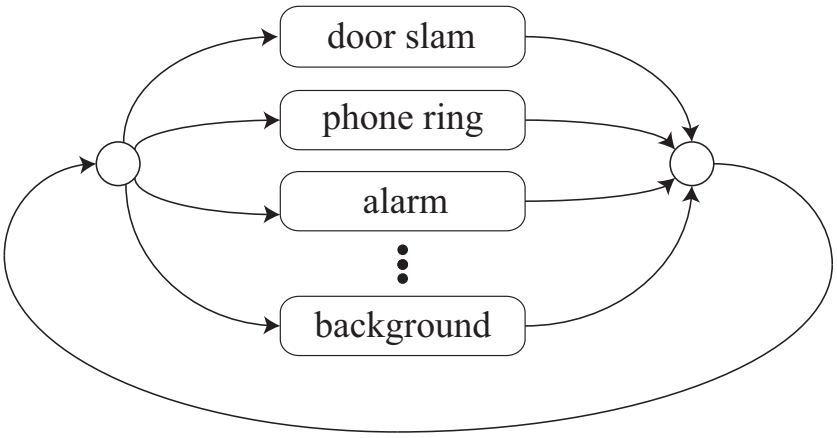

Figure 2.2: An exemplary network of HMM models for event detection (image adapted from [93]).

The left-to-right topology is the most commonly used one for HMM models [55, 144, 196, 245]. Although the ergodic HMM has also been studied [22, 144], it is more complex and has been shown to yield a performance comparable to the left-to-right one [144]. Other customized topologies have also been investigated. The CMU system submitted to the CLEAR 2006 challenge [216] exploited variant topologies for different event categories. These topologies were induced using the greedy k-variable k-means algorithm proposed by Reyes-Gomez and Ellis [186]. Niessen et al. [157] made use of the hierarchical HMM proposed in [113] in which the hidden state representation is split into two layers. The state variables in the upper layer represent sound events as usual whereas those in the lower layer are intended to represent the smaller units of the sound events. Due to the additional layer, this model aims at capturing the temporal dependencies between sub-events within an event class as well as between sound events. Zhuang et al. [244] proposed to use the tandem connectionist-HMM [95]. The problem is that in a typical HMM system each hidden state models only local observations. The tandem architecture alleviates this issue by using a neural network to model a larger temporal context around the local features and, therefore, allows one to gain event discrimination. The posterior probabilities outputted by the network will then augment the HMM in state sequence modeling.

Typically, one HMM model is used for each event category. The number of emitting states usually ranges from one to five [22, 93, 144]. The observation distributions of the states are commonly modeled by Gaussian mixtures. The number of Gaussians needed to model each state varies from two up to 128 [22, 216,

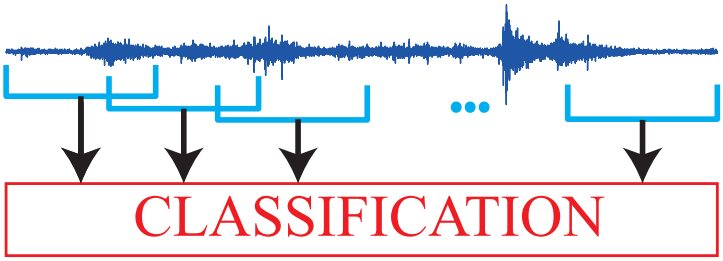

Figure 2.3: Detection-by-classification approach for audio event detection in a sliding window fashion.

245]. Moreover, an additional single-state HMM is typically used as an universal background model. Putting them altogether, the event HMMs as well as the background HMM are connected into a network as in Figure 2.2. The network can be interpreted as a high-level fully connected HMM in which each state is an event. Therefore, the whole system is actually a cascaded HMM. Eventually, audio events are jointly segmented and classified from an audio stream via the Viterbi decoding algorithm [67, 182].

2.2.2 Detection-by-Classification Approach

In general, detection systems adhering to the detection-by-classification approach encompass two classifiers: (1) one is trained to distinguish the target events from background, (2) the other to classify the target events into different categories of interest. These learned classifiers are employed to detect audio events in audio streams in a sliding window fashion as demonstrated in Figure 2.3. At each time, the foreground/background classifier is exercised first to reject background segments. The event segments are subsequently classified and assigned class labels by the event classifier. To achieve that, the audio segments typically need to be long enough (one second for example) in order to capture sufficient signal statistics, so that they can be recognized individually.

While this approach is straightforward, it is also benefited from the diversity of audio representations (cf. Section 2.1) and classifiers. Many classification algorithms have been attempted, such as Support Vector Machine (SVM) [85, 180, 195, 212, 213], GMM [143, 243], HMM [12, 157], k-nearest-neighbors (kNN) [88], Adaboost [129], random forest classification [60, 157, 202], and template matching [13, 114, 179]. Moreover, inspired by the success of deep learning, the community recently encountered an influx of works using deep learning techniques, especially in the

most recent DCASE 2016 evaluation campaign [1, 143]. For instance, Deep Neural Networks (DNNs) have been employed in [26, 37, 115], Convolutional Neural Networks (CNNs) in [61, 84, 121], and Recurrent Neural Networks (RNNs) in [2, 90, 163, 229, 248]. In order to leverage the advantages of different algorithms, combinations of classifiers of different types have also been studied [90, 122, 223, 244].

In the ASR-based approach, short noise in individual audio segments or frames can be mitigated by an HMM. That is because the HMM tends to learn self-transitions in the hidden finite state machine. Unfortunately, there exists no such mechanism in case of the detection-by-classification approach. Obtained segment-wise class label sequences are usually noisy. As a result, they require to be smoothed with a post-processing step, for instance moving average filtering [230], median filtering [215], dilation filtering [77], majority voting [22], or using an HMM [71]. In general the ASR-based approach is more efficient than the detection-by-classification counterpart in the detection task [214, 242].

2.3 Audio Event Classification (AEC)

Opposing to the detection task which is to detect event instances in continuous streams, the classification task aims at assigning a class label to every isolated event instance as illustrated in Figure 2.4. That is, one assumes a perfect segmentation of the event instances from the continuous signals. Basically, this task seeks to answer how the event instances of a particular category can be separated from those of other categories of interest.

Due to the intra- and inter-class duration variation, the AEC task needs to deal with variable-length audio signals since they cannot just simply be scaled to an equal length. The conventional approach is to form a global feature vector for each audio event by aggregating statistics over frame-level low-level features [48, 139, 214]. More commonly, higher-level global representations are learned automatically on top of low-level features, such as bag-of-words representations [160, 180]. All the features presented in Section 2.1 can be applied here. After the feature extraction stage, the event classification problem can be fed into any classification framework. The common classification algorithms, such as kNN [139], GMM [39, 139], HMM [22, 47, 214], SVM [160, 180, 214, 221], have been widely used for this purpose.

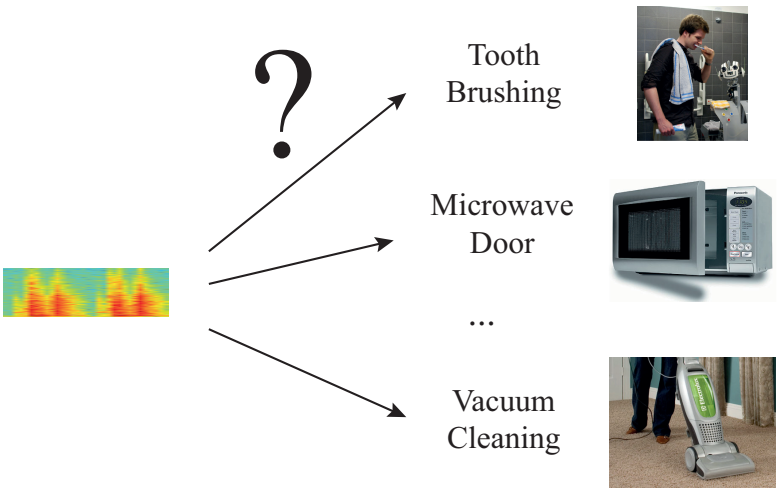

Figure 2.4: Illustration of audio event classification.

Recently, deep learning methods, mostly DNNs [97, 140] and CNNs [178, 211, 240], have drawn significant attention for this task. However, a typical DNN or CNN is not able to handle variable-length inputs properly. A work-around is to decompose the events into multiple equal segments. The DNNs and CNNs are then trained for segment-wise classification. The segment-wise decisions are finally aggregated into the final one using majority voting [140, 240] or probabilistic voting [97, 178, 211].

2.4 Temporal Coding

In general, the characteristics of audio events differ from those of speech in the manner that they cover a much wider variety in frequency content and duration. However, they are similar in one perspective. Speech exposes temporal structure, i.e. it is possible to decompose words into their constituent phones. Likewise, an audio event can be decomposed into atomic units of sound [32, 116]. For example, the sound of a "using water tap" event may further be decomposed into the sounds of the water running in the faucet, then pushing into the air, and finally splashing into the sink. As a result, the patterns of unit sequences can be used as a signature to distinguish different sound events. Therefore, aggregating temporal configurations of audio events is a promising approach for the detection and classification tasks. However, unlike phones in speech, it is not easy to design or discover the sound unit dictionary to encode all sound events.

There have been several attempts trying to capture event structural information for detection and classification. In ASR, HMMs have been proven efficient in sequential modeling. Therefore, ASR-based detection systems [55, 93, 144] can be seen as the direct adaptation of ASR techniques to the AED task. However, the common assumption of first-order Markov process makes these HMMs suboptimal in capturing complex temporal structure of audio events. There are some recent works employing Long Short Term Memory networks (LSTMs) as alternative methods to model longer dependency between audio segments [163, 234]. Nevertheless, their improvements over the conventional HMMs were vague mainly due to the unavailability of the important ingredient, a sufficiently large dataset.

Alternatively, an audio event can be considered as a sequence of atomic units of sounds [32, 35, 116] and the pattern of occurrences is then used as an event signature. The sequence of frame-wise feature vectors can also be represented with pyramid BoW models [180] and extensions of BoW models with augmented temporal information [85, 161]. By considering the audio event as a constellation of local features, their arrangement can be encoded using Self-Organizing Maps (SOMs) [51], Hough transforms [50], or accumulation of local feature voting [205].

2.5 Handling Overlapping Events

Depending on target environments (such as kitchen rooms [205], bathrooms [33], car inside space [148], or meeting rooms [215, 242], etc.), a detection/classification system may experience overlapping events or events overlapped with speech which require tailored strategies to deal with. For the ASR-based approach, in order to cope with event overlap, one strategy is to conduct multiple passes of the Viterbi decoding over the test signal [55, 93, 144]. At each iteration, for every frame the decoding algorithm is prohibited from entering the states that have been occupied by the decoded paths in the previous iterations, except for the state of the background model. By this, the next-best path different from those in the previous iterations can be obtained. Another strategy is to adapt a multi-class detection problem into multiple one-against-all subproblems [22]. In a subsystem, beside the HMM model trained for a particular target category, another one is trained to model all the rest, i.e. background and other event classes. Detection on a test signal is then accomplished by multiple passes of binary detectors over

the signal. The decision sequences are eventually superimposed to yield the final decision.

In [213, 215], event overlaps are considered as a new class and was handled separately in detection-by-classification systems. When an event overlap is detected, the mixture of class labels is subsequently identified using a hierarchical clustering scheme. This method, however, requires training data of not only all possible class combinations but also different overlapping degrees that is not always available in practice.

A different class of methods is to untangle the event mixtures using source separation techniques. A typical system employs NMF on the time-frequency representation of the target signals to pre-process and learn a dictionary of sounds from either the isolated events or the event mixtures for template matching purpose [52, 54, 94, 145]. In [50], recognition of isolated overlapping events was treated as a feature selection problem in which the events are represented by a constellation of discriminative local time-frequency patches. The arrangement of these patches was further modeled by a Hough transform. More recently, Cakir et al. [26] showed that mixtures of sounds can be classified directly using multi-label DNNs. This scheme was further improved in [163], in which the authors employed multi-label RNNs in the form of bi-directional long short-term memory (BLSTM) in replacement of the multi-label DNNs. An advantage of the RNNs is that they are able to model sequential information directly and, therefore, avoid postprocessing as in [26].

2.6 Multi-Channel and Multi-Modal Fusion

The majority of works in literature have tackled the single-microphone problem mainly due to its simplicity. However, multiple channels [76, 120, 213] or multiple modalities [24, 25, 27, 104], when available, provide different views on the same problem. Fusion of multiple data sources, therefore, can help to improve the spatial coverage or to gain robustness to event overlaps. Multi-channel fusion has been studied in [76, 108, 120, 215, 246]. A certain event can happen in any location, for example in a meeting room, which is not known in advance. Therefore, there is no preference for a placement location of a single microphone. The fusion systems were intended to leverage the spatial information of distributed microphones to compensate for low signal-to-noise (SNR) events. Due to the fact that acoustic sources do not overlap in video modality, visual features extracted from videos are

used to account for overlapping events [23, 27, 104]. Acoustic localization features have also been found useful for this purpose [22, 24].

Three fusion strategies have been widely used: early fusion [24, 27, 76], intermediate fusion [104, 108, 120], and late fusion [76, 104, 213, 215]. The first includes multi-channel training [76], signal combination with a beamformer [76], plain feature concatenation [24, 27], and feature weighting [104]. The second either processes the concatenated features in the early fusion method with an additional transform [108] or considers the posterior outputs from different channel-wise classifiers as intermediate features [104, 120]. Lastly, the third combines channel-wise decisions into the final decision with majority voting [213], maximum probability voting [76], or posterior multiplicative combination [76], to possibly retain the reliabilities of each data channel or modality.

2.7 Weak Labeling vs. Strong Labeling

There are applications, such as multimedia indexing [96, 118] and birdsong classification [20, 207], in which a weak annotation is sufficient, meaning that the annotation only indicates which audio events occur in an audio recording, and their temporal information is not included. This is mainly due to the scope of these applications. People are only interested in whether or not a certain event happened in a recording, and the exact time is not needed. However, it is much more common in the field that the target audio events are strongly annotated, i.e. their onset and offset time in a continuous audio signal are specified. This setting has been used throughout the international challenges so far [143, 204, 208, 216]. Furthermore, it will be shown in this study that the temporal segmentation of the target events is useful for reducing false alarms outputted by a detection system, leading to improvements on overall detection performance.

2.8 Open Issues

There exist several important open issues that this study is going to investigate. First, while event temporal structure is important, most of the previous works have succeeded in modeling them for classification of isolated events [50, 51, 85, 180]. Attempts to use the classifiers for the detection task have resulted in significant deterioration of their capability. This is likely due to the mismatch

between training and test data caused by the fixed sliding window length of the detection-by-classification approach. On the other hand, the HMMs in the ASR-based detection approach are limited in capturing long dependency between audio segments due to their assumption of the first-order Markov process [55, 93, 144]. Although RNNs can overcome this limitation and are potential alternatives for sequence modeling in general, the community still lacks sufficiently large datasets to harness them. It is therefore worth investigating other temporal structure coding schemes for both classification and detection tasks. They should be capable of capturing long-term dependencies of audio segments while making the data mismatch in the detection-by-classification approach irrelevant.

Second, for the detection task, beside the majority of works focusing on improving overall performance in terms of detection accuracy, other aspects of the problem have also been studied, such as overlapping event handling [26, 50, 163, 220] and multiple-channel fusion [76, 120]. However, little attention has been paid to two important aspects: early detection and false positive reduction. The ability of early event detection is important for many applications, especially safety-related ones. In intuition, a temporal audio event has duration and its audio samples arrive at a detection system one-by-one. The more data the system receives, the better it knows and gains confidence about the event. As a result, when the system accumulates enough confidence about the target event, it can be certainly detected even if only partial event duration is seen.

The third issue is false positive reduction. In fact, it has been inherently addressed when improving the overall performance of a detection system towards an oracle one, which makes no mistakes. However, while reaching the accuracy level of the oracle system is difficult if not impossible, the question is whether one can achieve the goal of false positive reduction given the state-of-the-art performance of the detection system. Proper removal of false alarms would help to improve the overall detection performance.

Finally, concerning audio event representation, although considerable progress has been made and many features have been proposed for different benchmark datasets, these representations are derived based on analysis of target event signals per se. Comparing to these data-specific learned features, the hand-crafted ones, e.g. MFCCs, are generic in the sense that the feature extraction process is the same for different datasets. On contrary, the learned features that have been proposed so far are data-specific. They are induced to be as much discriminative as possible

for the specific audio events under analysis. In terms of feature learning, the community is still in need for a way to automatically learn generic descriptors for audio events. Such a generic representation would be very useful for dealing with different tasks at hand in a homogeneous way as the hand-crafted features do.

3 Audio Event Detection with Random Regression Forests

This chapter presents the proposed AED system. The AED problem is posed as a regression problem which is then addressed using the random forest regression framework [43, 69]. This approach therefore differs from the majority of contributions in the field which rely on the detection-by-classification and ASR-based approaches. In fact, it opens up a third path to tackle the problem.

A regression forest is an ensemble of decision trees, each of which plays the role of a nonlinear mapping from an input space into a continuous output space. Each tree is constructed in such a way that the original problem is divided into smaller ones, which are solvable with simple models. A split node in the tree maintains a test that is applied to a data sample to direct it towards the child nodes. The tests are optimized by some criteria to recursively group the training samples into clusters. A good prediction can then be achieved by simple models at leaf nodes. These models are computed using training samples which reach the leaves during the tree construction process. While overfitting likely happens for a decision tree alone, an ensemble of randomly trained trees enjoys high generalization [72].

Motivated by the success of random regression forests in various computer vision tasks [43, 69, 185], they are adapted for the AED task in this chapter. The proposed approach relies on the fact that many audio events possess strong temporal structures. As a result, when an event is decomposed into multiple segments, the segments can be used to infer the event onset and offset positions. The idea is, therefore, to model these temporal structures for event detection. To accomplish this, the isolated training examples of a target event category are firstly divided into multiple segments. Each segment is represented by a feature vector and associated with a two-dimensional distance vector. The entries of the distance vector store the distances from the segment to the corresponding event onset and offset. A regression forest is then trained to group the audio segments into clusters at the leaf nodes of its trees so that those segments in the same cluster

have similar onset and offset distances. It turns out that these distances can be modeled with Gaussian distributions. During the testing phase, being presented with an unseen audio segment, each tree of the trained forest maps the segment to one of its formed clusters. Estimations for the onset and offset distances of the test segment are then obtained by the corresponding Gaussian distributions. The onset and offset positions of the target event in an audio stream can finally be inferred. As long as the onset and offset positions are found, the target event is certainly detected and its temporal boundary is also determined.

3.1 Random Regression Forests for Event Onset and Offset Estimation

In this section, the details of the algorithm that is used to train a random regression forest will be described, followed by the estimation procedure in which the trained regression forest is employed to estimate the onset and offset distances of a test audio segment. Finally, the inference step that is used to estimate the onset and offset positions of the target event in an audio stream will be elaborated.

3.1.1 Training Random Regression Forests

The objective of the training algorithm is to train a random regression forest to model the temporal structure of a target event category. The training algorithm is based on the random regression forest framework [43].

3.1.1.1 Training data

Each annotated event instance in the training data is decomposed into a sequence of audio segments. Each segment is represented by a column vector $\mathbf{x} \in \mathbb{R}^D$ of D low-level features (the employed features will be described in detail in Section 3.3.3). In addition, the segment is also associated with a distance vector $\mathbf{d} = \begin{pmatrix} d^+ & d^- \end{pmatrix}^\mathsf{T} \in \mathbb{R}^2_+$. The entries d^+ and d^- denote the distances from the segment to the corresponding event onset and offset, respectively. They will also be referred to as the onset and offset distances from now on. Assume that the segment is at the time index n, d^+ and d^- are then computed as:

$$d^+ = n - n^+, \tag{3.1}$$

$$d^- = n^- - n. \tag{3.2}$$

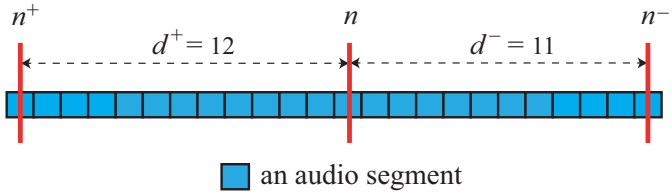

Figure 3.1: The onset and offset distance of the audio segment at the time index n to the event onset n^+ and the event offset n^-.

In Eqs. (3.1) and (3.2), n^+ and n^- denote the time indices of the first segment (e.g. the onset) and the last segment (e.g. the offset) of the event, respectively. An example of the onset and offset distances is illustrated in Figure 3.1.

Eventually, an audio segment set $\mathcal{S}_{\text{train}} = \{(\mathbf{x}_i, \mathbf{d}_i);\ i = 1, 2, \ldots, N_{\text{train}}\}$ of size N_{train} is obtained from all training event instances. This set will be used to train the regression forest.

3.1.1.2 Training algorithm

For convenience, let us define the following quantities related to a certain audio segment set $\mathcal{S} \neq \emptyset$:

$$x_{r,\min}(\mathcal{S}) = \min_{(\mathbf{x},\mathbf{d})\in\mathcal{S}} x_r, \tag{3.3}$$

$$x_{r,\max}(\mathcal{S}) = \max_{(\mathbf{x},\mathbf{d})\in\mathcal{S}} x_r, \tag{3.4}$$

$$\bar{d}^+(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{x},\mathbf{d})\in\mathcal{S}} d^+, \tag{3.5}$$

$$\bar{d}^-(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{x},\mathbf{d})\in\mathcal{S}} d^-, \tag{3.6}$$

$$\Sigma^+(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{x},\mathbf{d})\in\mathcal{S}} \left(d^+ - \bar{d}^+(\mathcal{S})\right)^2, \tag{3.7}$$

$$\Sigma^-(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{x},\mathbf{d})\in\mathcal{S}} \left(d^- - \bar{d}^-(\mathcal{S})\right)^2, \tag{3.8}$$

$$V(\mathcal{S}) = \sum_{(\mathbf{x},\mathbf{d})\in\mathcal{S}} \left\|\mathbf{d} - \bar{\mathbf{d}}\,(\mathcal{S})\right\|_2^2. \tag{3.9}$$

In Eqs. (3.3) and (3.4), the variable x_r denotes the value of the feature vector $\mathbf{x}$ at the feature channel $r \in \{1, 2, \ldots, D\}$. The quantities $x_{r,\min}(\mathcal{S})$ and $x_{r,\max}(\mathcal{S})$ are the minimum and maximum values of the feature channel r with respect to

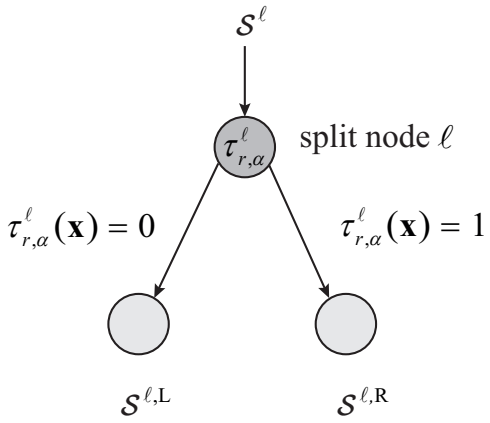

Figure 3.2: A binary test $\tau^{\ell}_{r,\alpha}$ at the split node ℓ, applied on the audio segments of the set $\mathcal{S}^{\ell}$ to divide it into two subsets: $\mathcal{S}^{\ell,\mathrm{R}}$ on the right and $\mathcal{S}^{\ell,\mathrm{L}}$ on the left.

the audio segments in $\mathcal{S}$, respectively. In Eqs. (3.5), (3.6), (3.7), and (3.8), the variables $\bar{d}^{+}(\mathcal{S})$, $\bar{d}^{-}(\mathcal{S})$, $\Sigma^{+}(\mathcal{S})$, and $\Sigma^{-}(\mathcal{S})$ represent the mean onset distance, the mean offset distance, the onset distance variance, and the offset distance variance with respect to the audio segments, respectively. Finally, in Eq. (3.14), $V(\mathcal{S})$ denotes the *distance variation* of the audio segments where $\bar{\mathbf{d}}\,(\mathcal{S})$ indicates their mean distance vector and is given by

$$\bar{\mathbf{d}}\,(\mathcal{S}) \equiv \begin{pmatrix} \bar{d}^{+}(\mathcal{S}) & \bar{d}^{-}(\mathcal{S}) \end{pmatrix}^{\mathsf{T}}. \tag{3.10}$$

Suppose that one wants to train a regression forest model $\mathcal{F} = \{\mathcal{T}_i;\ i = 1, 2, \ldots, N_{\mathcal{F}}\}$ where $\mathcal{T}_i$ denotes the i-th decision tree of the forest and $N_{\mathcal{F}}$ is the number of trees. In order to construct a tree $\mathcal{T} \in \mathcal{F}$, a segment subset $\mathcal{S}^{\mathcal{T}} \subset \mathcal{S}_{\text{train}}$ is randomly sampled from $\mathcal{S}_{\text{train}}$ and used for the training purpose. The tree is grown recursively, starting from the root node to the split nodes, and finally to the leaf nodes. Note that the root node is a split node, but does not have a parent node. Without loss of generality, let us consider a current split node ℓ and let $\mathcal{S}^{\ell} \subset \mathcal{S}^{\mathcal{T}}$ denote the segment subset associated with the node ℓ (the root node is therefore linked with the subset $\mathcal{S}^{\mathcal{T}}$). At this node, a pool of binary tests is randomly generated. These tests have the form given by

$$\tau^{\ell}_{r,\alpha}(\mathbf{x}) = \begin{cases} 1, & \text{if } \ x_r > \alpha \\ 0, & \text{otherwise.} \end{cases} \tag{3.11}$$

Here, the feature channel r is randomly selected in the set $\{1, 2, \ldots, D\}$ and the variable α denotes a random threshold generated in the range $[x_{r,\min}(\mathcal{S}^\ell), x_{r,\max}(\mathcal{S}^\ell)]$. Evaluating a test $\tau^\ell_{r,\alpha}$ on the segments of the set $\mathcal{S}^\ell$ will split them into two subsets $\mathcal{S}^{\ell,\mathrm{R}}$ and $\mathcal{S}^{\ell,\mathrm{L}}$:

$$\mathcal{S}^{\ell,\mathrm{R}} = \left\{ (\mathbf{x}, \mathbf{d}) \in \mathcal{S}^\ell \,\middle|\, \tau^\ell_{r,\alpha}(\mathbf{x}) = 1 \right\}, \tag{3.12}$$

$$\mathcal{S}^{\ell,\mathrm{L}} = \left\{ (\mathbf{x}, \mathbf{d}) \in \mathcal{S}^\ell \,\middle|\, \tau^\ell_{r,\alpha}(\mathbf{x}) = 0 \right\}. \tag{3.13}$$

The splitting procedure is demonstrated in Figure 3.2. Hereafter, the optimal test $\tau^\ell_{r,\alpha,*}$ is adopted from the test pool to minimize the *total distance variation* $\left(V(\mathcal{S}^{\ell,R}) + V(\mathcal{S}^{\ell,L})\right)$:

$$\tau^\ell_{r,\alpha,*} = \arg\min_{\tau^\ell_{r,\alpha}} \left(V(\mathcal{S}^{\ell,\mathrm{R}}) + V(\mathcal{S}^{\ell,\mathrm{L}}) \right). \tag{3.14}$$

By doing this, the audio segments of the set $\mathcal{S}^\ell$ have been clustered based on both their features and their relative positions to the corresponding event onsets and offsets. Afterwards, $\mathcal{S}^{\ell,\mathrm{R}}$ and $\mathcal{S}^{\ell,\mathrm{L}}$ are further sent to the right and left child nodes of the split node ℓ, accordingly.

The splitting process is repeated recursively until either a maximum depth D_{stop} of the tree is reached or a minimum number N_{stop} of audio segments are left. When at least one of the conditions is met, a leaf node is created. Let $\mathcal{S}^{\mathrm{leaf}} \subset \mathcal{S}^{\mathcal{T}}$ denote the audio segment set reaching this leaf node. The mean distance vector $\bar{\mathbf{d}}\left(\mathcal{S}^{\mathrm{leaf}}\right)$ and the distance covariance matrix $\mathbf{\Sigma}\left(\mathcal{S}^{\mathrm{leaf}}\right)$ of the audio segments in $\mathcal{S}^{\mathrm{leaf}}$ are then computed and stored at the leaf. The $\bar{\mathbf{d}}\left(\mathcal{S}^{\mathrm{leaf}}\right)$ is calculated as in Eq. (3.10) and the distance covariance matrix $\mathbf{\Sigma}\left(\mathcal{S}^{\mathrm{leaf}}\right)$ is defined as

$$\mathbf{\Sigma}\left(\mathcal{S}^{\mathrm{leaf}}\right) = \begin{pmatrix} \Sigma^+\left(\mathcal{S}^{\mathrm{leaf}}\right) & 0 \\ 0 & \Sigma^-\left(\mathcal{S}^{\mathrm{leaf}}\right) \end{pmatrix}, \tag{3.15}$$

where $\Sigma^+(\cdot)$ and $\Sigma^-(\cdot)$ are given in Eqs (3.7) and (3.8), respectively. In addition, the distance vectors of the audio segments in $\mathcal{S}^{\mathrm{leaf}}$ are modeled as a two-dimensional Gaussian distribution $\mathcal{N}\left(\mathbf{d} \,\middle|\, \bar{\mathbf{d}}\left(\mathcal{S}^{\mathrm{leaf}}\right), \mathbf{\Sigma}\left(\mathcal{S}^{\mathrm{leaf}}\right)\right)$ where

$$\mathcal{N}\left(\mathbf{d} \,\middle|\, \bar{\mathbf{d}}, \mathbf{\Sigma}\right) = \frac{1}{2\pi\sqrt{\det(\mathbf{\Sigma})}} \exp\left(-\frac{1}{2} \left(\mathbf{d} - \bar{\mathbf{d}}\right)^\intercal \mathbf{\Sigma}^{-1} \left(\mathbf{d} - \bar{\mathbf{d}}\right) \right). \tag{3.16}$$

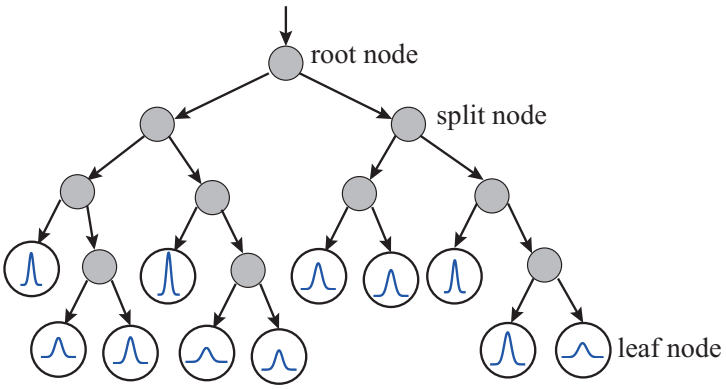

Figure 3.3: Illustration of a simplified regression tree.

Note that, for simplicity, the covariances between the onset and offset distances are not taken into consideration in the distance covariance matrix $\mathbf{\Sigma}(\mathcal{S}^{\text{leaf}})$ as defined in Eq. (3.15). The multi-variate Gaussian distribution $\mathcal{N}\left(\mathbf{d}\,\middle|\,\bar{\mathbf{d}}\left(\mathcal{S}^{\text{leaf}}\right), \mathbf{\Sigma}\left(\mathcal{S}^{\text{leaf}}\right)\right)$ can, therfore, be explicitly reduced into two univariate Gaussian distributions $\mathcal{N}^{+}\left(d^{+}\,\middle|\,\bar{d^{+}}\left(\mathcal{S}^{\text{leaf}}\right), \Sigma^{+}\left(\mathcal{S}^{\text{leaf}}\right)\right)$ and $\mathcal{N}^{-}\left(d^{-}\,\middle|\,\bar{d^{-}}\left(\mathcal{S}^{\text{leaf}}\right), \Sigma^{-}\left(\mathcal{S}^{\text{leaf}}\right)\right)$ where

$$\mathcal{N}\left(d\,\middle|\,\bar{d}, \Sigma\right) = \frac{1}{\sqrt{2\pi\Sigma}} \exp\left(-\frac{\left(d-\bar{d}\right)^{2}}{2\Sigma}\right). \tag{3.17}$$

These two Gaussian distributions separately model the onset and offset distances of the audio segments, respectively.

After training, the split nodes of the tree remain associated with the corresponding optimal tests, whereas its leaf nodes store the parameters of the Gaussian distributions. Figure 3.3 illustrates a simplified version of such a regression tree. The algorithm is repeatedly applied to grow all the trees in the forest $\mathcal{F}$. Moreover, since the learning algorithm is independent for every tree, they can be constructed in parallel.

3.1.2 Onset and Offset Distance Estimation

Given a test audio segment represented by the feature vector $\mathbf{x} \in \mathbb{R}^{D}$, the learned regression forest $\mathcal{F}$ is employed to estimate the onset and offset distances of this test segment. The decision trees of the forest are firstly used to perform individual estimations which are then averaged to yield the global estimation of the whole forest.

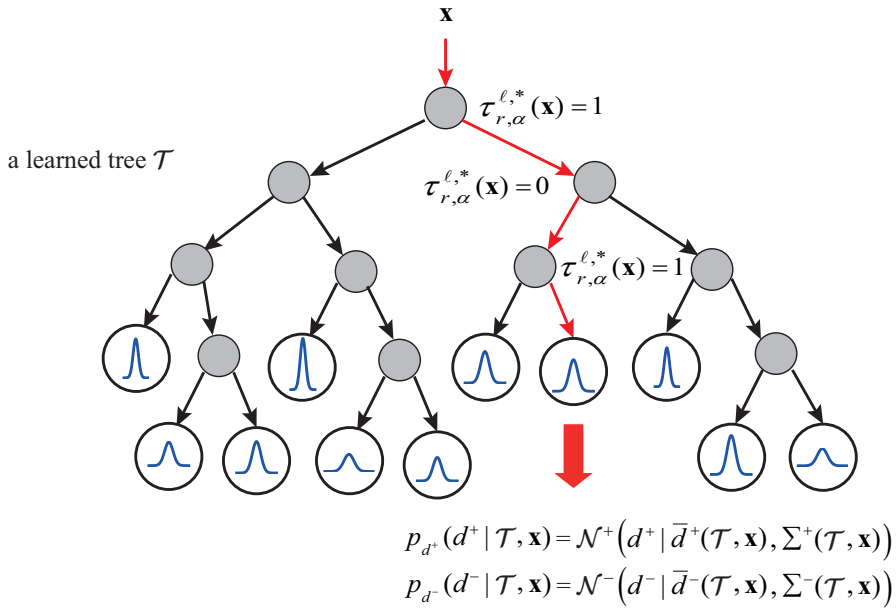

Figure 3.4: Illustration of the testing procedure of a regression tree $\mathcal{T}$ given a test audio segment represented by the feature vector $\mathbf{x}$.

The testing procedure of a learned tree $\mathcal{T} \in \mathcal{F}$ is demonstrated in Figure 3.4. Being presented with the test audio segment, the tree $\mathcal{T}$ will direct it down from the root to a leaf node at the bottom. At each split node along the path, the optimal binary test stored at the node is exercised on $\mathbf{x}$. The audio segment is then forwarded either to the right child node if the test output is 1 or to the left child node if the test output is 0. These steps are repeatedly carried out until the audio segment ends up at the final leaf node.

Let $\bar{d}^+(\mathcal{T},\mathbf{x})$, $\Sigma^+(\mathcal{T},\mathbf{x})$, $\bar{d}^-(\mathcal{T},\mathbf{x})$, and $\Sigma^-(\mathcal{T},\mathbf{x})$ denote the mean onset distance, the mean offset distance, the onset distance variance, and the offset distance variance stored at the final leaf node of the tree $\mathcal{T}$ given the test audio segment $\mathbf{x}$, respectively. The onset and offset distance estimates of the test audio segment are obtained by the following probability density functions:

$$p_{d^+}(d^+ \,|\, \mathcal{T},\mathbf{x}) = \mathcal{N}^+\left(d^+ \,|\, \bar{d}^+(\mathcal{T},\mathbf{x}), \Sigma^+(\mathcal{T},\mathbf{x})\right), \tag{3.18}$$

$$p_{d^-}(d^- \,|\, \mathcal{T},\mathbf{x}) = \mathcal{N}^-\left(d^- \,|\, \bar{d}^-(\mathcal{T},\mathbf{x}), \Sigma^-(\mathcal{T},\mathbf{x})\right), \tag{3.19}$$

respectively. The global estimation made by the whole forest $\mathcal{F}$ is then computed by averaging the individual estimations produced by its trees:

$$p_{d^+}\left(d^+ \,\middle|\, \mathbf{x}\right) = \frac{1}{|\mathcal{F}|}\sum_{\mathcal{T}\in\mathcal{F}} \mathcal{N}^+\left(d^+ \,\middle|\, \bar{d}^+\left(\mathcal{T},\mathbf{x}\right), \Sigma^+\left(\mathcal{T},\mathbf{x}\right)\right), \tag{3.20}$$

$$p_{d^-}\left(d^- \,\middle|\, \mathbf{x}\right) = \frac{1}{|\mathcal{F}|}\sum_{\mathcal{T}\in\mathcal{F}} \mathcal{N}^-\left(d^- \,\middle|\, \bar{d}^-\left(\mathcal{T},\mathbf{x}\right), \Sigma^-\left(\mathcal{T},\mathbf{x}\right)\right). \tag{3.21}$$

The modes of $p_{d^+}\left(d^+ \,\middle|\, \mathbf{x}\right)$ and $p_{d^-}\left(d^- \,\middle|\, \mathbf{x}\right)$ indicate the onset and offset distances of the test audio segment estimated by the regression forest $\mathcal{F}$, respectively. To improve computational efficiency, the Gaussian distributions at a leaf node can be pre-computed in the distance ranges of the audio segments arriving at it during training and stored there. As a result, the regression step involves only scalar comparisons and additions.

3.1.3 Inference for Event Onset and Offset

The ultimate goal of the AED task is to determine the positions of possible event onsets and offsets in an audio stream. To accomplish this goal, the test signal is firstly divided into a sequence of segments as similarly done in the training step. From Eqs. (3.20) and (3.21), the onset and offset distance estimates of the audio segment $\mathbf{x}_m$ at the index m are given by:

$$p_{d^+}\left(d^+ \,\middle|\, \mathbf{x}_m\right) = \frac{1}{|\mathcal{F}|}\sum_{\mathcal{T}\in\mathcal{F}} \mathcal{N}^+\left(d^+ \,\middle|\, \bar{d}^+\left(\mathcal{T},\mathbf{x}_m\right), \Sigma^+\left(\mathcal{T},\mathbf{x}_m\right)\right), \tag{3.22}$$

$$p_{d^-}\left(d^- \,\middle|\, \mathbf{x}_m\right) = \frac{1}{|\mathcal{F}|}\sum_{\mathcal{T}\in\mathcal{F}} \mathcal{N}^-\left(d^- \,\middle|\, \bar{d}^-\left(\mathcal{T},\mathbf{x}_m\right), \Sigma^-\left(\mathcal{T},\mathbf{x}_m\right)\right). \tag{3.23}$$

Inserting Eq. (3.1) into Eq. (3.22) and Eq. (3.2) into Eq. (3.23), the estimates for the target event onset and offset positions read

$$p^+(n \,|\, \mathbf{x}_m) = \frac{1}{|\mathcal{F}|}\sum_{\mathcal{T}\in\mathcal{F}} \mathcal{N}^+\left(n \,\middle|\, m - \bar{d}^+\left(\mathcal{T},\mathbf{x}_m\right), \Sigma^+\left(\mathcal{T},\mathbf{x}_m\right)\right), \tag{3.24}$$

$$p^-(n \,|\, \mathbf{x}_m) = \frac{1}{|\mathcal{F}|}\sum_{\mathcal{T}\in\mathcal{F}} \mathcal{N}^-\left(n \,\middle|\, m + \bar{d}^-\left(\mathcal{T},\mathbf{x}_m\right), \Sigma^-\left(\mathcal{T},\mathbf{x}_m\right)\right), \tag{3.25}$$

respectively. The interpretation for Eqs. (3.24) and (3.25) is that each Gaussian distribution in Eq. (3.24) has been placed at $\bar{d}^+\left(\mathcal{T},\mathbf{x}_m\right)$ backward from m and each Gaussian distribution in Eq. (3.25) has been placed at $\bar{d}^-\left(\mathcal{T},\mathbf{x}_m\right)$ forward from m as illustrated in Figure 3.5.

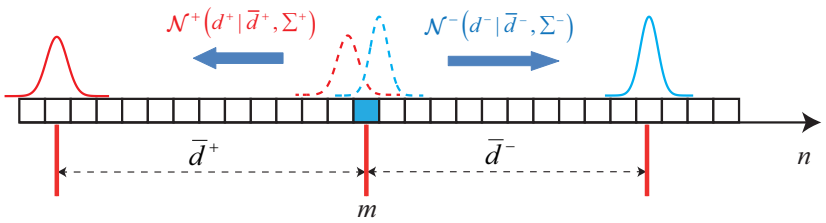

Figure 3.5: Illustration of inferring the event onset and offset positions by placing the Gaussian distribution $\mathcal{N}^+(d^+ \,|\, \bar{d}^+, \Sigma^+)$ backward at $\bar{d}^+$ and the Gaussian distribution $\mathcal{N}^-(d^- \,|\, \bar{d}^-, \Sigma^-)$ forward at $\bar{d}^-$ relative to the current time index m.

The estimations obtained by all audio segments are finally accumulated to yield the confidence scores, which indicate the occurrence likelihoods of the target event onset and offset:

$$f^+(n) = \sum_m p^+\left(n \,\middle|\, \mathbf{x}_m\right), \tag{3.26}$$

$$f^-(n) = \sum_m p^-\left(n \,\middle|\, \mathbf{x}_m\right), \tag{3.27}$$

respectively. Ideally, if there exists only one target event instance in the test signal, its onset and offset positions can be determined as

$$\hat{n}^+ = \arg\max_n f^+(n), \tag{3.28}$$

$$\hat{n}^- = \arg\max_n f^-(n), \tag{3.29}$$

respectively. As long as the onset and offset positions are found, the target event will be detected and its temporal boundaries will also be determined. In practice, an audio stream typically contains multiple event occurrences, one after another, which must be detected sequentially. In addition, multiple event types are usually targeted in the same system. The extensions to cope with these issues will be elaborated in the next section.

3.2 Multi-Class AED System

The proposed regression forests described in the previous section are specific for a single event category. In general, it is common in practical applications that multiple event types are targeted in the same system. Assume that there are C

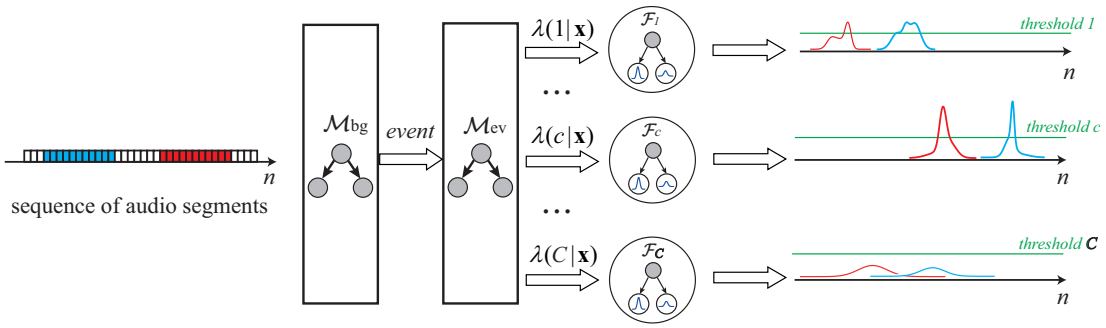

Figure 3.6: The pipeline of the multi-class audio event detection system.

such event categories in total. The architecture of the proposed multi-class event detection system is depicted in Figure 3.6.

For each event category $c \in \{1, 2, \ldots, C\}$, a class-specific regression forest $\mathcal{F}_c$ is learned using the algorithm in Section 3.1. In addition, the two following segment-wise classifiers are trained:

- $\mathcal{M}_{\text{bg}}$: the binary classifier to tell apart foreground segments from background ones.

- $\mathcal{M}_{\text{ev}}$: the multi-class event classifier that classifies an audio segment into one of the event categories of interest.

Both classifiers are trained using random-forest classification [18] which has been proven to be computationally efficient to deal with sufficiently large amount of data (for the dataset used in the experiment, the training and testing data contain 614,460 and 156,745 audio segments, respectively). Furthermore, random-forest classification naturally supports probability output that will be later utilized to weight the contribution of an audio segment to the event onset and offset estimations.

Regarding the processing pipeline, the audio segments of the test signal are firstly presented to the binary classifier $\mathcal{M}_{\text{bg}}$, which filters out the background and only allows event segments getting through. Subsequently, these event segments are classified by the multi-class event classifier $\mathcal{M}_{\text{ev}}$ to produce the posterior probabilities over the class labels. Finally, they are fed into the regression forests for estimating target event onset and offset positions.

Eqs. (3.24) and (3.25) can be adapted to yield onset and offset estimations for a target event of class c given the audio segment $\mathbf{x}_m$ at the time index m:

$$p^+(n, c \,|\, \mathbf{x}_m) = \boldsymbol{\lambda}(c \,|\, \mathbf{x}_m)\, p^+(n \,|\, \mathbf{x}_m, c)\,, \tag{3.30}$$

$$p^-(n, c \,|\, \mathbf{x}_m) = \boldsymbol{\lambda}(c \,|\, \mathbf{x}_m)\, p^-(n \,|\, \mathbf{x}_m, c)\,, \tag{3.31}$$

respectively. In Eqs. (3.30) and (3.31), the weighting function $\boldsymbol{\lambda}(c \,|\, \mathbf{x}_m)$ is used to regulate the contribution of the audio segment $\mathbf{x}_m$ into the estimations. Three following weighting schemes are studied:

1. **Weighting Scheme 1.** With this scheme, only audio segments that are classified as class c by the classifier $\mathcal{M}_{\text{ev}}$ are allowed to contribute to the estimations. Furthermore, their contributions are equally counted. The weighting function reads

$$\boldsymbol{\lambda}(c \,|\, \mathbf{x}_m) = \mathbb{I}(\hat{c} = c), \tag{3.32}$$

where the predicted class label $\hat{c}$ is computed as

$$\hat{c} = \underset{c \in \{1,2,\ldots,C\}}{\arg\max}\; P(c \,|\, \mathbf{x}_m)\,. \tag{3.33}$$

The indicator function $\mathbb{I}(\cdot)$ is given by

$$\mathbb{I}(x) = \begin{cases} 1 & \text{if} \quad x \text{ is } true \\ 0 & \text{if} \quad x \text{ is } false, \end{cases} \tag{3.34}$$

where $x \in \{true, false\}$. The classification posterior probability $P(c \,|\, \mathbf{x}_m)$ in Eq. (3.33) is modeled by the event classifier $\mathcal{M}_{\text{ev}}$.

2. **Weighting Scheme 2.** This scheme weights the contribution of an audio segment to the overall estimations by the posterior probability of it belonging to class c. The rationale is that audio segments recognized as class c with higher confidence than others should contribute more into the estimations and vice versa. The weighting function reads

$$\boldsymbol{\lambda}(c \,|\, \mathbf{x}_m) = \mathbb{I}(\hat{c} = c) P(c \,|\, \mathbf{x}_m)\,. \tag{3.35}$$

3. **Weighting Scheme 3.** While Weighting Scheme 1 and 2 enforce to count only contributions of those segments classified as class c, this scheme encourages every audio segment $\mathbf{x}_m$ with $P(c \,|\, \mathbf{x}_m) > 0$ to contribute to the estimations. The weighting function reads

$$\lambda(c \,|\, \mathbf{x}_m) = P\left(c \,|\, \mathbf{x}_m\right). \tag{3.36}$$

Using this scheme, the shared features between different event categories are taken into account at the cost of the need to exercise the regression forests more often.

Eventually, the estimates by all individual audio segments are summed up to produce the final onset and offset estimates for target events of class c:

$$f_c^+(n) = \sum_m p^+\left(n, c \,|\, \mathbf{x}_m\right), \tag{3.37}$$

$$f_c^-(n) = \sum_m p^+\left(n, c \,|\, \mathbf{x}_m\right). \tag{3.38}$$

That is, via the learned regression model, the local audio segments are used to vote for the boundaries of the target events. By doing this, the "shape" of an audio event, i.e. its temporal extent, is implicitly modeled as a constellation of its local segments [130] as illustrated in Figure 3.7. The higher the confidence scores $f_c^+(n)$ and $f_c^-(n)$ are at a time index, the more likely it is that there are event onset and offset occurring at it.

Typically, an audio stream will contain multiple sequential event instances of a certain class, resulting in multiple peaks in both onset and offset confidence scores $f_c^+(n)$ and $f_c^-(n)$. In addition, the scores are likely to be noisy, especially for event instances with low SNRs. However, the peaks are expected to be clear above the noise floor so that the events can be detected. Here, a thresholding method is employed to locate them.

Firstly, for normalization, the onset and offset confidence scores are divided by their respective maximum values on the training audio signals. These maximum values can be determined by cross-validation on the training audio signals. Thereafter, a class-specific detection threshold $\beta_c \in [0, 1]$ is commonly applied on both $f_c^+(n)$ and $f_c^-(n)$ to eliminate the noise below them. The peaks on $f_c^+(n)$ and $f_c^-(n)$ are finally determined as the maximum values above the threshold β_c. The idea is demonstrated in Figure 3.8 for three event categories of the ITC-Irst

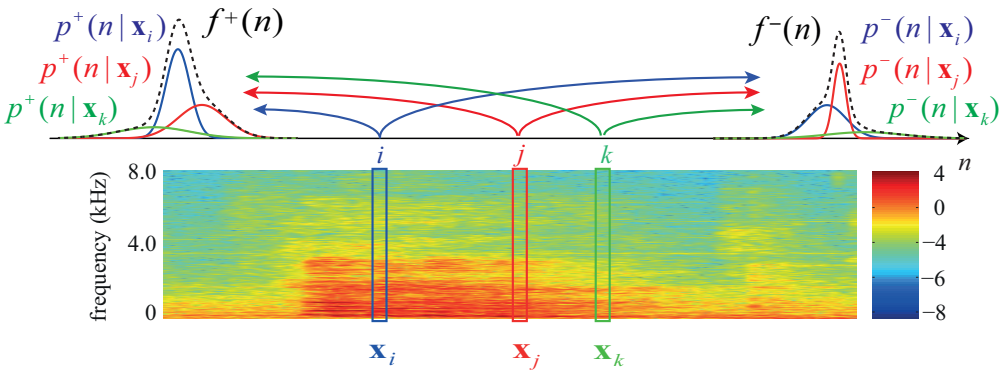

Figure 3.7: Illustration of event onset and offset estimation. The confidence scores $f^+(n)$ and $f^-(n)$ for the target event onset and offset estimations are computed by summing the individual estimates obtained by its audio segments (demonstrated here with three segments represented by $\mathbf{x}_i$, $\mathbf{x}_j$, and $\mathbf{x}_k$, respectively). Note that the class label is ignored here for simplicity.

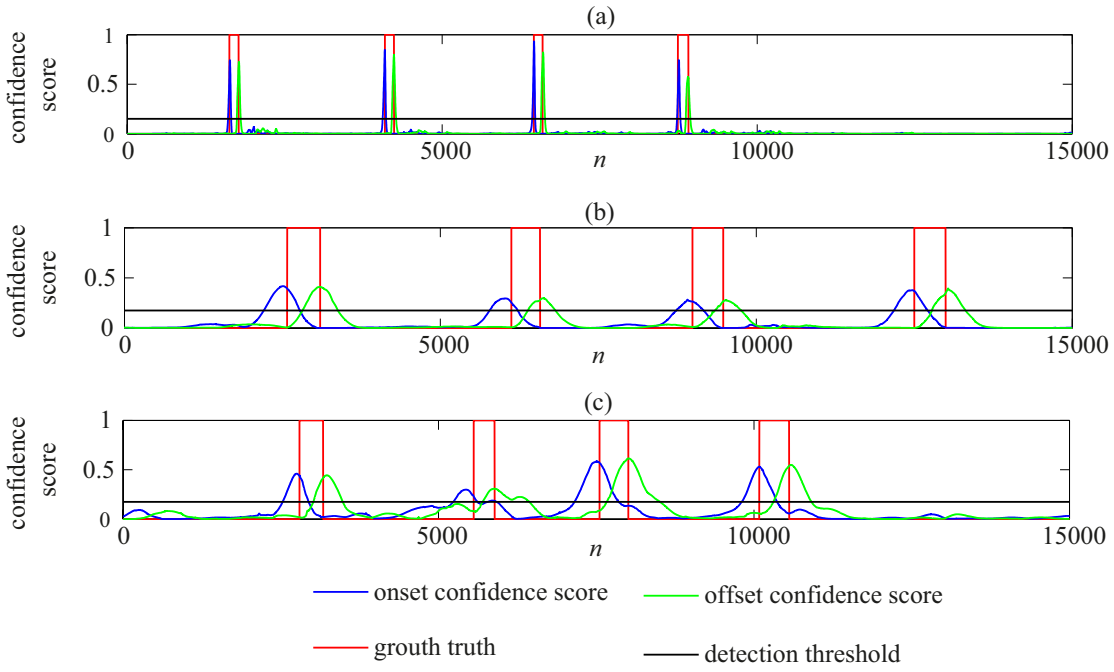

Figure 3.8: Illustration of applying class-specific detection thresholds to determine the peaks of onset and offset confidence scores for three selected categories of the ITC-Irst dataset: (a) “door slam”, (b) “spoon cup jingle”, and (c) “steps”.

database [214] (more details in Section 3.3). A pair of maximum scores, an onset peak followed by an offset peak, are considered as the boundary of a detected event. It is necessary to enforce a rule to couple two peaks whose values are closest to each other to overcome the case when there exists more than one such pair.

3.3 Experiments

This section presents the experiments conducted on the ITC-Irst dataset [214]. The experimental setup, including the employed dataset, the evaluation metrics, the employed low-level features, and the parameters related to the proposed detection algorithm, are first described. Afterwards, the detection performance of the proposed system is presented and compared with those of the two baseline detection systems. The influence of the number of trees in the regression forests on the detection performance will also analyzed.

3.3.1 Dataset

The experiments were conducted on the ITC-Irst dataset [214, 247] which was recorded in meeting-room environments. The recording room was equipped with 32 microphones. 28 of them were distributed in seven T-shaped arrays mounted on the walls and four other single microphones were mounted on a table as shown in Figure 3.9. The dataset consists of twelve recording sessions with a total duration of 1.7 hours. There also exists background noise due to different types of noise sources, such as PC fans and air-conditioning systems.

There are 16 semantic event categories with approximately 50 events recorded for most of the categories. A summary of the dataset is shown in Table 3.1. The dataset has been extensively examined in the CLEAR evaluations [204, 214]. To be consent with the CLEAR 2006 challenge [214], twelve classes are targeted for evaluation, including “door knock” (kn), “door slam” (ds), “steps” (st), “chair moving” (cm), “spoon cup jingle” (cl), “paper wrapping” (pw), “key jingle” (kj), “keyboard typing” (kt), “phone ring” (pr), “applause” (ap), “cough” (co), and “laugh” (la). The rest was considered as background. Many of the events are subtle (e.g. “steps”, “chair moving”, and “keyboard typing”), making the task more challenging. Nine recording sessions were used for training and the three remaining ones were used for testing as in the standard setting of the CLEAR 2006 challenge [214]. In these experiments, only one channel named *TABLE_1*

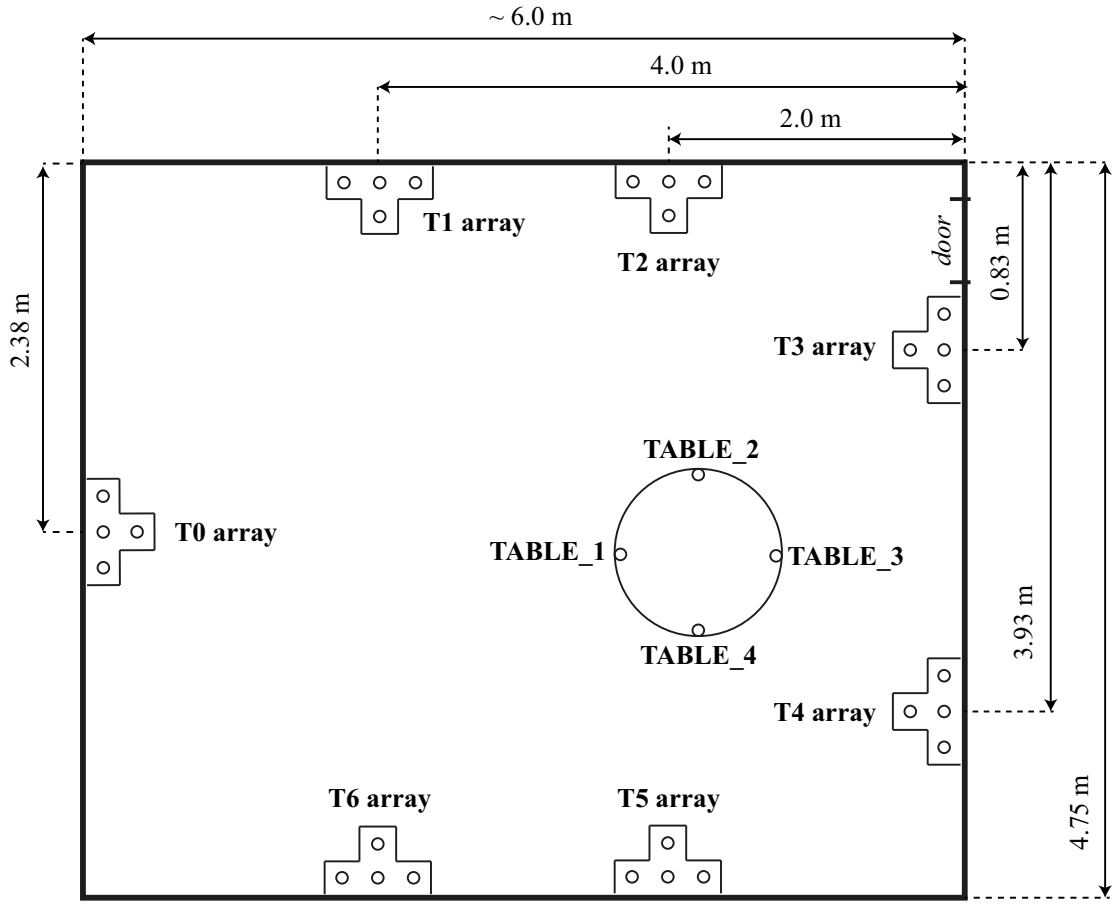

Figure 3.9: The recording room and microphone layout of the ITC-Irst dataset (image adapted from [246]).

positioned nearly at the center of the recording room [214, 247] was employed in the experiments.

3.3.2 Parameters

The classifiers $\mathcal{M}_{\text{bg}}$ and $\mathcal{M}_{\text{ev}}$ were trained with random-forest classification [18] with 200 trees each. The per-class regression forests were trained with the random regression forest algorithm described in Section 3.1 with ten trees each. For a category c, a randomly sampled subset containing 50% audio segments of the corresponding training set was used to train each tree of the regression forest $\mathcal{F}_c$. During training, 20,000 binary tests were randomly generated at a split node of a tree. In addition, the maximum tree depth and the minimum number of audio segments for early stopping were set to $D_{\text{stop}} = 12$ and $N_{\text{stop}} = 20$, respectively.

Table 3.1: Sumary of the ITC-Irst dataset [214].

Event category	**#event instances**		**#audio segments**	
	Training	**Testing**	**Training**	**Testing**
door knock	35	12	5,977	1,983
door slam	39	12	6,263	2,076
steps	38	12	17,866	4,810
chair moving	35	12	11,556	3,812
spoon cup jingle	36	12	21,989	7,065
paper wrapping	36	12	18,519	7,149
key jingle	36	12	23,655	8,421
keyboard typing	35	12	21,603	7,647
phone ring	66	23	38,824	12,316
applause	9	3	5,345	1,894
cough	36	12	7,233	3,046
laugh	36	12	7,003	2,459
door open	36	13	6,386	1,715
falling object	36	12	5,127	1,613
phone vibration	10	3	5,052	1,474
mimo pen buzz	36	12	24,144	9,528
unknown	17	9	2,647	2,033
Total	572	195	229,189	79,041

In order to determine the detection threshold for an event category, nine-fold leave-one-out cross validation was conducted over nine training audio recordings. Particularly, it was noticed that cross-validation training for the classifiers $\mathcal{M}_{\text{bg}}$ and $\mathcal{M}_{\text{ev}}$ is sufficient. The regression forests trained with the whole training data can be employed for both testing and cross-validation purposes. The class-specific detection thresholds were then searched in the range $[0, 1]$ with a step size of 0.05. Eventually, those threshold values which yield maximum class-specific F1-scores were chosen. The experiments were repeated five times and the average performance is reported here.

3.3.3 Employed Low-Level Features

The audio signals were downsampled to 16 kHz and decomposed into interleaved 100 ms long segments with an overlap of 90 ms. Although any low-level acoustic

features can be used to describe an audio segment, a set of very typical features was exploited: 16 log-frequency filter bank coefficients [152], their first and second derivatives, zero-crossing rate, short-time energy, four sub-band log energies, spectral centroid, and spectral bandwidth. The total number of features is 53.

Given the waveform $s(n)$ of length $L = 1600$ of an audio segment, the magnitude of the short-time Fourier transform was firstly computed using an F-point fast Fourier transform (FFT) with $F = 2048$:

$$S(k) = \left| \sum_{n=0}^{L-1} s(n) w_L(n) e^{\frac{-j2\pi nk}{F}} \right|, \tag{3.39}$$

where $k = 0, \ldots, \frac{F}{2} - 1$. The function $w_L(n)$ denotes the L-point Hamming window:

$$w_L(n) = 0.54 - 0.46 \cos(\frac{2\pi n}{L-1}), \tag{3.40}$$

where $0 \leq n \leq L - 1$. The employed features are computed as below.

1. *Zero-crossing rate (ZCR).* This feature indicates the number of zero crossings within the audio segment $s(n)$ and is given by

$$ZCR = \sum_{n=1}^{L-1} \mathbb{I}\Big(s(n)s(n-1) < 0\Big). \tag{3.41}$$

   The indicator function $\mathbb{I}(\cdot)$ is given in Eq. (3.34).

2. *Short-time energy (STE).* This feature represents the total energy of the audio segment $s(n)$. It is computed as

$$STE = \sum_{n=0}^{L-1} \left(s\left(n\right)\right)^2. \tag{3.42}$$

3. *Spectral centroid (SC).* This feature represents the spectral "brightness" of the spectrum. It is defined as the normalized weighting average frequency with the weights are the spectral magnitudes $S(k)$ given in Eq. (3.39):

$$SC = \frac{\sum_{k=0}^{\frac{F}{2}-1} \zeta(k) S(k)}{\sum_{k=0}^{\frac{F}{2}-1} S(k)}. \tag{3.43}$$

In the above equation, $\zeta(k) = \frac{2kf_s}{F}$ denotes the frequency at the index k, and f_s is the sampling frequency.

4. *Spectral bandwidth (SB).* This feature quantifies the spreading of the spectrum around the spectral centroid SC and is computed as

$$SB = \frac{\sum_{k=0}^{\frac{F}{2}-1} \left(\zeta\left(k\right) - SC\right)^2 S^2\left(k\right)}{\sum_{k=0}^{\frac{F}{2}-1} S^2(k)}. \tag{3.44}$$

5. *Log-frequency filter bank coefficients (FFBC).* In order to compute these features, the spectral magnitudes $S(k)$ were warped with 16 Mel-scale filter banks, followed by logarithm scaling. The l-th log-frequency filter bank coefficient, for $1 \leq l \leq 16$, is computed as

$$FFBC(l) = \log\left(\sum_{k=0}^{\frac{F}{2}-1} S(k) H_{\text{mel}}(l,k)\right), \tag{3.45}$$

where

$$H_{\text{mel}}(l,k) = \begin{cases} 0 & \text{if} \quad \zeta(k) < \mathfrak{f}(l-1) \\ \frac{\zeta(k)-\mathfrak{f}(l-1)}{\mathfrak{f}(l)-\mathfrak{f}(l-1)} & \text{if} \quad \mathfrak{f}(l-1) \leq \zeta(k) < \mathfrak{f}(l) \\ 1 & \text{if} \quad \zeta(k) = \mathfrak{f}(l) \\ \frac{\mathfrak{f}(l+1)-\zeta(k)}{\mathfrak{f}(l+1)-\mathfrak{f}(l)} & \text{if} \quad \mathfrak{f}(l) < \zeta(k) \leq \mathfrak{f}(l+1) \\ 0 & \text{if} \quad \zeta(k) > \mathfrak{f}(l-1), \end{cases} \tag{3.46}$$

and

$$\mathfrak{f}(l) = 700\left(\exp\left(\frac{l}{1125}\right) - 1\right). \tag{3.47}$$

The functions $H_{\text{mel}}(\cdot,\cdot)$ in Eq. (3.46) and $\mathfrak{f}(\cdot)$ in Eq. (3.47) represent the Mel-scale filter banks and the formula for converting from Mel scale to frequency. The *FFBCs* were further filtered with the filter $H(z) = z - z^{-1}$ for decorrelation and liftering purpose [152]. Afterwards, 15 first-order and 14 second-order derivatives of the *FFBCs* were computed in frequency direction and included into the feature set.

6. *Subband energies (SBE).* 16 computed *FFBCs* were divided into four non-overlapping equal subbands (i.e. each of them covers $F_{sub} = 4$ consecutive *FFBCs*). The energies were then calculated for each subband i as

$$SBE(i) = \sum_{l=(i-1)F_{sub}}^{iF_{sub}-1} FFBC(l) \tag{3.48}$$

for $1 \leq i \leq 4$.

3.3.4 Baseline Systems

The two following detection baseline systems were implemented for comparison purpose:

- **SVM**: This system conforms to the common detection-by-classification scheme. It uses a sliding window of one second and a shift of 100 ms on audio signals for detection. The detection task is accomplished by two nonlinear SVM classifiers with the radial basis function (RBF) kernel given by

$$\kappa(\mathbf{x}, \mathbf{z}) = \exp(-\gamma||\mathbf{x} - \mathbf{z}||^2), \tag{3.49}$$

where $\mathbf{x}$ and $\mathbf{z}$ denote two feature vectors. One of the classifiers is used for event/background discrimination and the other is for subsequent event classification. The *libSVM* library [31] was used for training and testing. Grid search was performed with 10-fold cross-validation to search for the hyperparameters of the SVMs. For the hyperparameter C that trades off errors of the SVMs on training data and margin maximization, a coarse grid search was firstly conducted in the set $\left\{2^k; k \in \{-2, -1, \ldots, 8\}\right\}$. It was then followed by a fine search in the set $\left\{2^k; k \in \{C_* - 1, C_* - 0.75, \ldots, C_* + 1\}\right\}$ where C_* is the optimal value found by the previous coarse search. A similar grid search was also conducted for the parameter γ of the RBF kernel given in Eq. (3.49). For representation, each one-second segment was further decomposed into 25 ms frames with an overlap of 50%. A set of 53 low-level features as described in Section 3.3.3 was then extracted for a frame. In turn, a global feature vector which consists of mean and standard deviation of the per-frame feature vectors was used to represent the one-second segments. Furthermore, a median filter of size 17 was applied on the label sequences to

eliminate too short silences or non-silences. This setting is similar to that of the *UPC-D* system of the CLEAR 2006 evaluation [214].

- **HMM**: This system complies with the ASR framework. The employed features and parameters are similar to those of the winning *ITC-D1* system in the CLEAR 2006 evaluation [214]. The audio signals were divided into short 20 ms audio frames with a hop size of 10 ms as commonly used for speech. MFCCs were calculated for each frame with a Hamming window and 24 Mel bands. Beside the first 13 coefficients (including 0-th coefficients), 13 delta coefficients, and 13 acceleration coefficients were also calculated using a window length of five frames. Each event category was described by a three-state HMM. All the HMMs have a left-to-right topology and use output probability densities represented by mixtures of 32 Gaussian components with diagonal covariance matrices. HMM training was accomplished through the Baum-Welch training procedure [235]. Finally, the optimum event sequence was obtained by the Viterbi decoding algorithm [182]. The *HTK toolkit* [238] was employed for training and testing.

3.3.5 Evaluation Metrics

Following the CLEAR 2006/2007 challenges [214, 215], two evaluation metrics were used for evaluation: the detection error rate (ER) and the detection F1-score. A ground-truth event is considered to be mapped as long as there exists at least one event hypothesis whose center falls inside its interval. A ground-truth event is considered correctly detected if it is mapped by an event hypothesis and their labels are matched. Then, the ER metric is computed as

$$\mathrm{ER} = \frac{N_{\mathrm{del}} + N_{\mathrm{ins}} + N_{\mathrm{sub}}}{N}, \tag{3.50}$$

where

- N : the number of ground-truth events,
- N_{del} : the number of unmapped ground-truth events,
- N_{ins} : the number of unmapped event hypotheses,
- N_{sub} : the number of mapped event hypotheses with mismatched class labels.

In another words, N_{del}, N_{ins}, and N_{sub} denote the deletion error, insertion error, and substitution error, respectively. Note that ER may exceed 100% because of the additional insertion errors.

The F1-score measure is defined as

$$\text{F1-score} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \tag{3.51}$$

where

$$\text{precision} = \frac{\text{the number of correct event hypotheses}}{\text{the number of event hypotheses}}, \tag{3.52}$$

$$\text{recall} = \frac{\text{the number of correctly detected ground-truth events}}{\text{the number of ground-truth events}}. \tag{3.53}$$

3.3.6 Experimental Results

This section elaborates on the performance of the proposed detection system on the experimental dataset and its performance variation as a function of the number of trees in the forests. A performance comparison between the proposed system and the baseline systems will also be presented.

3.3.6.1 Overall detection performance and comparison

The class-specific detection thresholds found by cross-validation are shown in Figure 3.10. These thresholds reflect characteristics of the confidence scores of different event categories. The large threshold of a category (e.g. "chair moving") implies a high noise floor in its confidence scores, which is mostly caused by the high ambiguity between audio segments of this class and both background ones and those of other categories. In contrast, a low threshold indicates a low noise level in confidence scores and a low such ambiguity (e.g. "phone ring").

To emphasize the importance of the regression models in the detection pipeline in Figure 3.6, the classification accuracies of the two segment-wise classifiers $\mathcal{M}_{\text{bg}}$ and $\mathcal{M}_{\text{ev}}$ are firstly examined. With independent testing, their accuracies are 87.6% and 76.6%, respectively. If they are evaluated sequentially, $\mathcal{M}_{\text{bg}}$ followed by $\mathcal{M}_{\text{ev}}$, the accuracy of the $\mathcal{M}_{\text{ev}}$ declines to 69.3% since the wrongly classified segments made by $\mathcal{M}_{\text{bg}}$ are transferred to the next step. In addition, there are on average 23.9% of false-positive segments, causing noise in segment-wise label sequences.

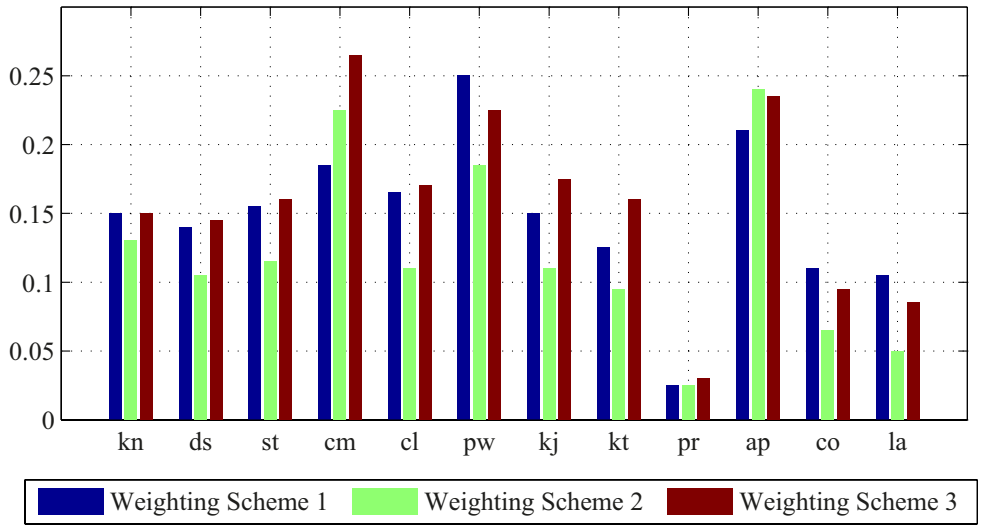

Figure 3.10: Class-wise detection thresholds β which were found to maximize the cross-validation F1-score.

Therefore, the successor regression models can also be thought of as post-processing operators that connect the correct-classified segments and overwrite the spurious ones to produce homogeneous segment sequences of audio events.

The overall detection results of different systems are shown in Tables 3.2 and 3.3 for the two evaluation metrics ER and F1-score, respectively. Among three weighting schemes of the proposed system, Weighting Scheme 1 yields the lowest performance whereas Weighting Scheme 3 is the best one. By considering the hard classification label, it is likely that Weighting Scheme 1 experiences quantization errors. On the contrary, utilizing the classification posterior probabilities as weights, Weighting Scheme 2 is able to mitigate this effect, improving, although incrementally, both ER and F1-score by 0.1% absolute. Significant performance improvements are obtained by Weighting Scheme 3, 3.0% absolute on ER and 1.1% absolute on F1-score, compared to Weighting Scheme 2. It underlines the importance of taking into account the shared features between different classes.

For a comprehensive comparison, Table 3.4 shows overall detection performances of the proposed system with different weighting schemes, the baselines, and the CLEAR 2006 submissions [214]. Compared to the baseline systems, the proposed system (Weighting Scheme 3) is able to reduce the ER metric by 15.7% and 23.9% absolute in comparison with the **SVM** and **HMM** baselines, respectively. Moreover, absolute gains of 9.4% and 8.7% can also be seen on the F1-score metric.

Table 3.2: ER (%) obtained by different detection systems. It is marked in bold where the proposed systems perform equally to or better than both the baselines.

Event Type	**SVM**	**HMM**	**Our systems**		
			Weighting Scheme 1	Weighting Scheme 2	Weighting Scheme 3
kn	0.0	16.7	8.3	6.7	1.7
ds	41.7	125.0	**5.0**	**0.0**	**0.0**
st	16.7	91.7	23.3	28.3	**15.0**
cm	58.3	91.7	**58.3**	**48.3**	**48.3**
cl	25.0	16.7	**3.3**	**10.0**	**3.3**
pw	8.3	16.7	**0.0**	**0.0**	**0.0**
kj	16.7	25.0	**8.3**	**16.7**	**5.0**
kt	33.3	25.0	33.3	35.0	41.7
pr	95.7	26.1	**26.1**	**26.1**	**23.5**
ap	0.0	0.0	**0.0**	**0.0**	**0.0**
co	33.3	25.0	**25.0**	**21.7**	**16.7**
la	58.3	41.7	**16.7**	**16.7**	**16.7**
Overall	30.8	39.0	**18.2**	**18.1**	**15.1**

Table 3.3: F1-score (%) obtained by different detection systems. It is marked in bold where the proposed systems perform equally to or better than both the baselines.

Event Type	**SVM**	**HMM**	**Our systems**		
			Weighting Scheme 1	Weighting Scheme 2	Weighting Scheme 3
kn	100.0	92.3	96.0	96.8	99.2
ds	96.3	69.6	**97.7**	**100.0**	**100.0**
st	92.0	69.6	88.7	86.3	**92.5**
cm	82.8	40.0	79.9	**83.6**	**84.4**
cl	85.7	92.3	**98.3**	**95.0**	**98.3**
pw	95.7	91.7	**100.0**	**100.0**	**100.0**
kj	91.7	87.0	**96.2**	**92.3**	**97.7**
kt	85.4	87.0	84.6	84.1	81.5
pr	67.6	93.2	90.5	90.5	89.9
ap	100.0	100.0	**100.0**	**100.0**	**100.0**
co	80.0	88.0	87.7	**92.0**	**92.2**
la	63.2	87.0	**90.9**	**90.9**	**90.9**
Overall	83.7	84.4	**91.9**	**92.0**	**93.1**

Table 3.4: Comparison of overall detection performance.

		ER	F1-score
Our systems	Weighting Scheme 1	18.2	91.9
	Weighting Scheme 2	18.1	92.0
	Weighting Scheme 3	15.1	93.1
Baselines	**SVM**	30.8	83.7
	HMM	39.0	84.4
CLEAR 2006	UPC-D	64.6	–
	CMU-D1	45.2	–
	ITC-D1	23.6	–

Overall, the proposed system yields better F1-score than both the baselines on nine out of twelve target categories as shown in Table 3.3. Finally, it also obtains lower ER than that of the winning CLEAR 2006 submission (i.e. ITC-D1 [214]) by 8.5% absolute.

For illustration purposes, Figure 3.11 shows the alignment of the detection results against the ground-truths on one of the test recordings for three systems: **HMM**, **SVM**, and the proposed system with Weighting Scheme 3. As can be seen from the figure, Weighting Scheme 3 system produces much less errors than the other two.

3.3.6.2 Varying the number of trees

It is well-known that the performance of the decision forests heavily depends on the number of the weak learners, i.e. the constituent trees [18, 43, 72]. It is understandable since the more trees are included, the better the *variance* of the final ensemble model can be reduced, leading to improvement of the generalization. The experiment in this section is conducted to study how the detection performance changes with the number of trees in the regression forest models. The number of

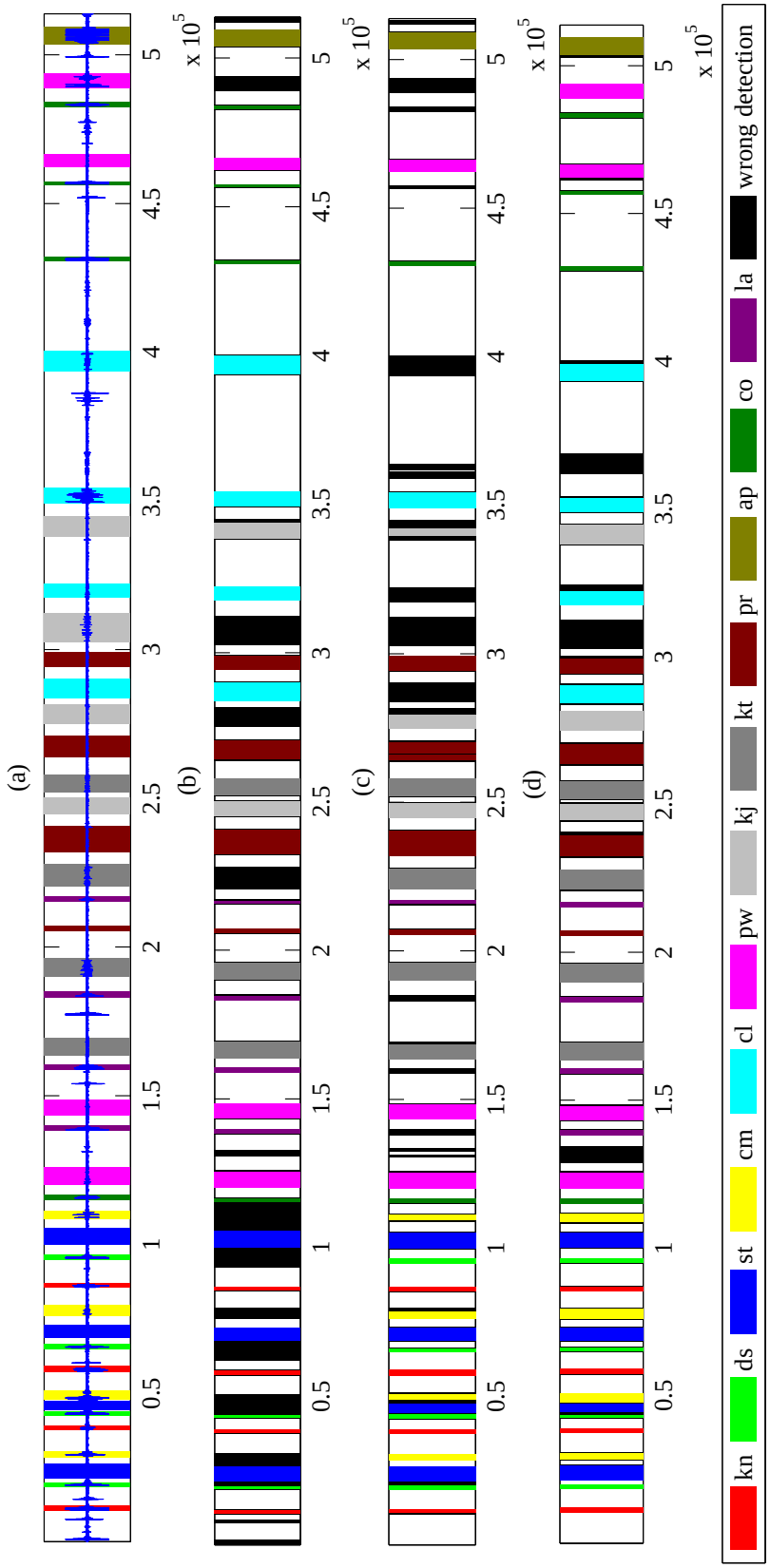

Figure 3.11: Ground-truths and detection outputs made by different systems on the first test audio file: (a) the ground-truths, (b) the outputs of **HMM**, (c) the outputs of **SVM**, and (d) the outputs of the proposed system with Weighting Scheme 3.

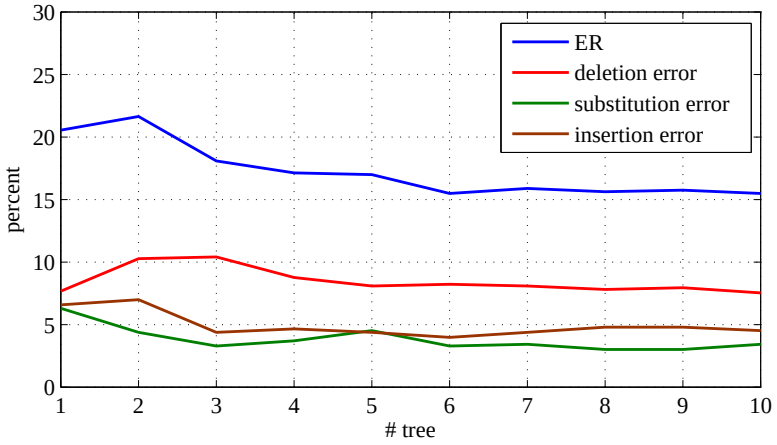

Figure 3.12: Variations of the overall ER, deletion error, insertion error, and substitution error as functions of the number of trees.

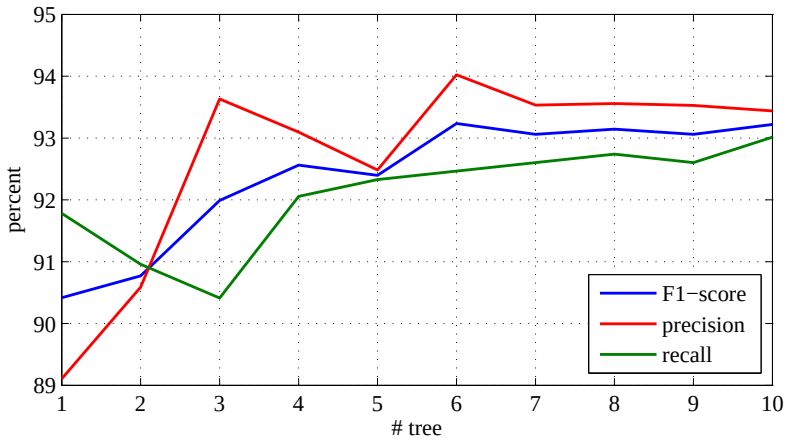

Figure 3.13: Variations of the overall F1-score, precision, and recall as functions of the number of trees.

regression trees was varied from one to ten. Note that, each time, the procedure of detection threshold search was repeated for a fair comparison.

The fluctuations of the overall detection error rate as well as its constituent errors (i.e. deletion, insertion, and substitution) are shown in Figure 3.12 as functions of the number of trees. As expected, they exhibit a long-term reducing trend as the number of tree increases. On the contrary, the overall F1-score, precision, and recall exhibit an increasing tendency as shown in Figure 3.13 in which they escalate with the increase of the number of trees.

4 Early Detection and Multi-Channel Fusion

In this chapter two important aspects of an audio event detection system, namely early detection and multi-channel fusion, will be investigated. The objective of the former is to detect the target events as soon as possible without losing the system's overall performance. It requires the ability to recognize a partial event as accurately as recognizing the entire event. This is achievable if the system's detection function holds the monotonicity property. The proposed system based on regression forests in Chapter 3 will be proven to meet this condition. The latter aims at leveraging spatial information of available distributed microphones to improve performance of the detection task. A simple, yet efficient fusion framework is proposed for multi-channel fusion. In this framework, the fusion system additively assembles confidence scores of per-channel regression forest detectors to gain confidence about occurrence of a target event.

4.1 Early Event Detection in Audio Streams

This section firstly expresses the problem of early event detection in audio streams which requires the monotonicity property of a detection function. The detection function of the proposed detection system presented in Chapter 3 is then proved to hold this property. Finally, the early detection ability of the proposed detection system is empirically verified via the experimental results on the ITC-Irst dataset [214].

4.1.1 Early Event Detection

Since a temporal event has duration, early detection, as in [99], means to detect the event as soon as possible, after it starts but before it ends. This idea is illustrated in Figure 4.1. In many situations, the early detection of the target

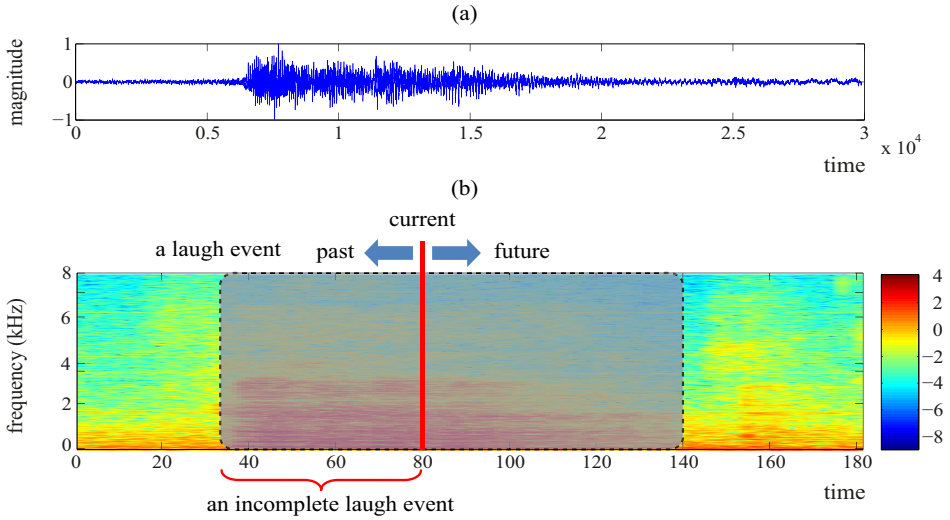

Figure 4.1: Can a "laugh" event be detected before it finishes? The "laugh" event is shown in (a) waveform and (b) spectrogram.

events is crucial, because without it, the intended applications would fail. As in the example from [99], if one wants to build a robot that interacts with humans, reliable and rapid event detection is a key requirement so that the robot can make appropriate responses in a timely manner. Otherwise, the responses would be out of synchronization. For another application in which a camera surveillance system is guided by an audio event detection system [156], the system needs to detect the events and take actions as quickly as possible. Directing the cameras after the events were already completed maybe too late, as the objects already moved. In general, the earliness of a detection system without losing its overall accuracy is always preferable. Thus, the early detection ability is an important property to guarantee the quality-of-service, especially for safety-related applications.

Despite the importance of early detection, little attention has been paid in prior works. Recently, a few works were explicitly proposed to address this problem. However, they are targeted for other modalities, e.g. videos in the field of computer vision [99, 102], rather than audios. So far, most of the proposed methods for AED, if not all, have focused on analyzing and detecting complete events. Since the detectors based on these methods are usually trained to recognize complete events only, they require observing the entire events for a reliable decision. Consequently, using them in an online mode for early detection, which requires the ability to recognize a partial event, would result in unreliable decisions due to the mismatch between the training and test data. More importantly, there is a reason that makes

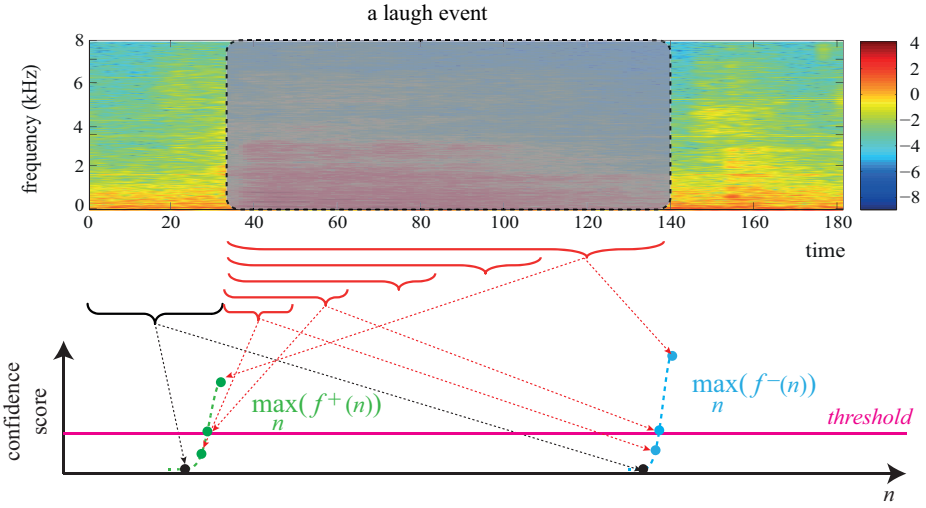

Figure 4.2: When both $\max_n (f^+(n))$ and $\max_n (f^+(n))$ reach the detection threshold, the target event is considered detected.

early detection difficult. Reliable early detection requires a monotonically growing detection function [99] that is not easy to obtain with these methods.

4.1.2 The Monotonicity of the Detection Function

It will be shown in this section that the proposed AED system based on random regression forests accommodates sequential data very well, enabling early detection. The monotonicity of the system's detection function can be further proven mathematically. More specifically, the maxima of detection onset and offset confidence scores in Eqs. (3.37) and (3.38) will either increase or remain unchanged as long as more audio segments are observed. While this property is essential for reliable early detection, it cannot be assured by a naive solution that simply detects a partial event. In addition, the proposed system also provides an efficient mechanism for event tracking. Although the structured output SVM framework proposed in [99] for early video event detection can be adapted for early AED, the formulation based on decision forests offers numerous advantages. First, it is unnecessary to augment the training process with partial events which cause exponential growth of the training data size. Second, there is no necessity to perform searching on multiple temporal scales for detection. Last but not least, the monotonicity of the scoring function in [99] is possibly no longer valid for periodic events, which is common for audio events. However, it is not the case in the proposed formulation.

Without loss of generality, let us assume a sequence of audio segments starting at m_1 and ending at m_4. For simplicity, the sequence is assumed to contain only one target event starting at m_2 and ending at m_3, where $m_1 < m_2 \leq m_3 < m_4$. Furthermore, let $\max_n (f^+(n))$ denote the maximum onset confidence score accumulated up to the time $\bar{m}$, where $m_1 \leq \bar{m} \leq m_4$ and $f^+(n)$ is given in Eq. (3.37). The class label is skipped here for simplicity. Due to Eq. (3.37), one has

$$\max_n \left(f^+(n) \right) = \max_n \Big(\sum_{m=m_1}^{\bar{m}} p^+(n \,|\, \mathbf{x}_m) \Big). \tag{4.1}$$

The position corresponding to $\max_n (f^+(n))$ is the estimated event onset position up to $\bar{m}$. Firstly, it is easy to show the strict monotonicity property of $\max_n (f^+(n))$, i.e. $\max_n (f^+(n)) < \max_n (f^+(n))$, on the event duration $[m_2, m_3]$ as below

$$\begin{aligned}
\max_n \left(f^+(n) \right) &= \max_n \Big(\sum_{m=m_1}^{\bar{m}} p^+(n \,|\, \mathbf{x}_m) \Big) \\
&< \max_n \Big(\sum_{m=m_1}^{\bar{m}} p^+(n \,|\, \mathbf{x}_m) \Big) + \min_n p^+(n \,|\, \mathbf{x}_{\bar{m}+1}) \\
&< \max_n \Big(\sum_{m=m_1}^{\bar{m}} p^+(n \,|\, \mathbf{x}_m) + p^+(n \,|\, \mathbf{x}_{\bar{m}+1}) \Big) \\
&= \max_n \Big(\sum_{m=m_1}^{\bar{m}+1} p^+(n \,|\, \mathbf{x}_m) \Big) \\
&= \max_n \left(f^+(n) \right).
\end{aligned} \tag{4.2}$$

The above strict monotonicity property is guaranteed since the constituent Gaussian distributions of $p^+(n \,|\, \mathbf{x})$ given in Eq. (3.24) have infinite support and are positive. Inspecting different disjoint segments of $[m_1, m_4]$, the monotonicity of $\max_n (f^+(n))$ over the whole $[m_1, m_4]$ can be proven:

$$\max_n \left(f^+(n) \right) = \max_n \left(f^+(n) \right) = 0 \text{ for } m_1 \leq \bar{m} < m_2 - 1, \tag{4.3}$$

$$\max_n \left(f^+(n) \right) < \max_n \left(f^+(n) \right) \text{ for } m_2 - 1 \leq \bar{m} \leq m_3 - 1, \tag{4.4}$$

$$\max_n \left(f^+(n) \right) = \max_n \left(f^+(n) \right) = \max_n \left(f^+(n) \right) \text{ for } m_3 \leq \bar{m} < m_4. \tag{4.5}$$

The proof for the offset confidence score $f^{-}(n)$ can also be derived similarly. The monotonicity can be interpreted as follows: the more the detector knows about the target event, the higher confidence it gains about the event occurrence.

Now, the question is how many audio segments are needed to accumulate adequate confidence scores and trigger an event reliably. The simple solution is that as soon as both peaks of accumulating confidence scores, $\max_n (f^{+}(n))$ and $\max_n (f^{-}(n))$, reach the pre-determined detection threshold β, the event is considered detected as illustrated in Figure 4.2.

4.1.3 Experiments on Early Audio Event Detection

To empirically verify the early detection ability of the proposed regression forest detector, the experiments in Section 3.3 are repeated on a simulated online setting. The proposed system with Weighting Scheme 3 is used here. A test audio stream is simulated as a sequence of audio segments which come to the system sequentially one-by-one. As a new event segment is available, the detection performance is evaluated again and recorded. The offline performance in Section 3.3 is used as the baseline. It will be shown that the events in the test signals can be detected correctly by the system even when their partial durations are observed. The online system offers the same performance as the offline system, but the events are detected much earlier.

Figure 4.3 illustrates how the online detection ER and F1-score develop as functions of the number of observed event segments. For all event classes, as more audio segments are observed, the online F1-scores keep increasing while the online ERs keep decreasing until they reach the offline system's ER and F1-score lines. More interestingly, it can also be seen in the figure that the online F1-score and ER curves always reach the offline ones before the maximum length of the events. That is, target events detectable by the system are always detected before they finish, although the earliness varies for different event types. For instance, from the F1-score curve of the "laugh" category, about 50% of events are correctly detected when approximately 75 segments (about 0.75 seconds) are seen. After that, the curve continues escalating as more and more audio segments are observed. The online curve reaches the offline F1-score baseline of 90.9% after observing about 140 segments (equivalent to 1.4 seconds). Considering that the "laugh" events last for approximately 400 segments, the online system only needs 35% of the event intervals to achieve the same detection accuracy as the offline system.

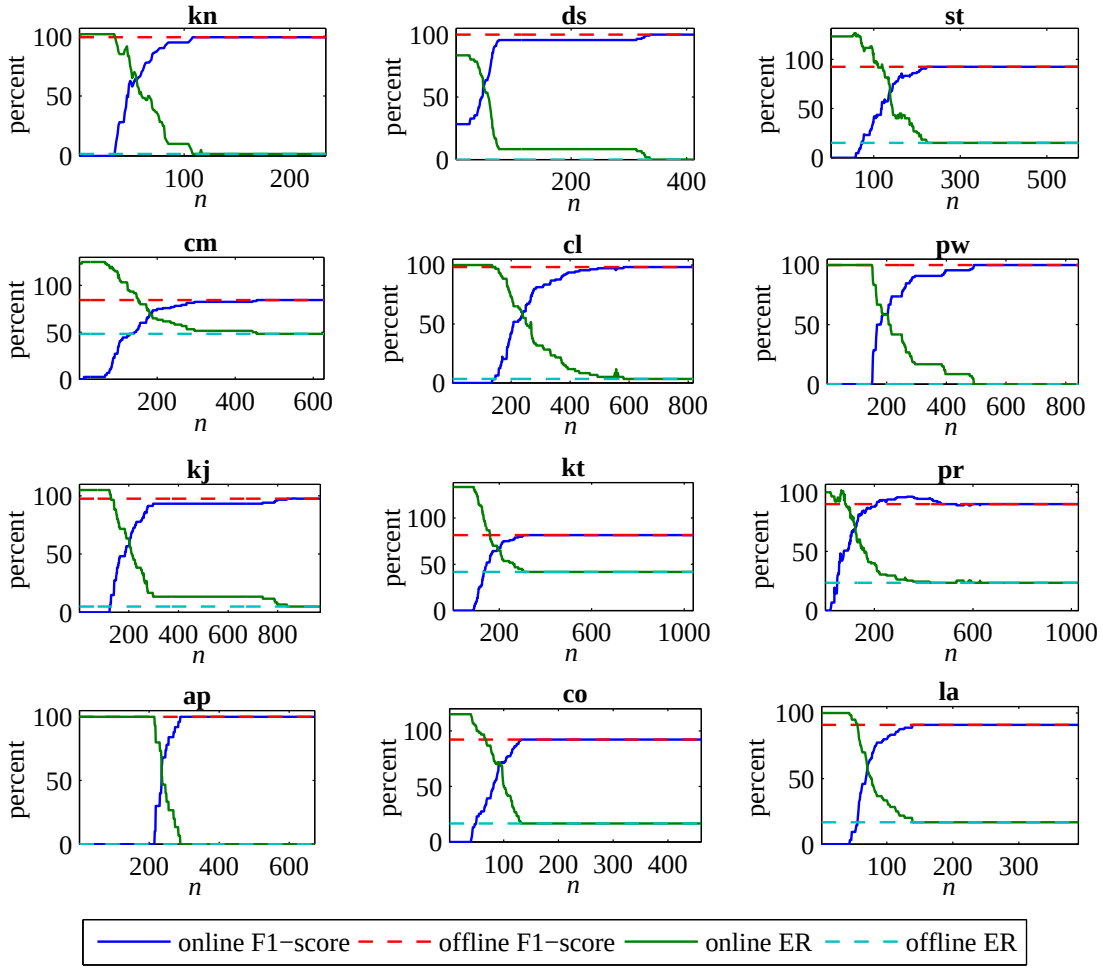

Figure 4.3: Online event detection results on different event classes as functions of the number of observed audio segments.

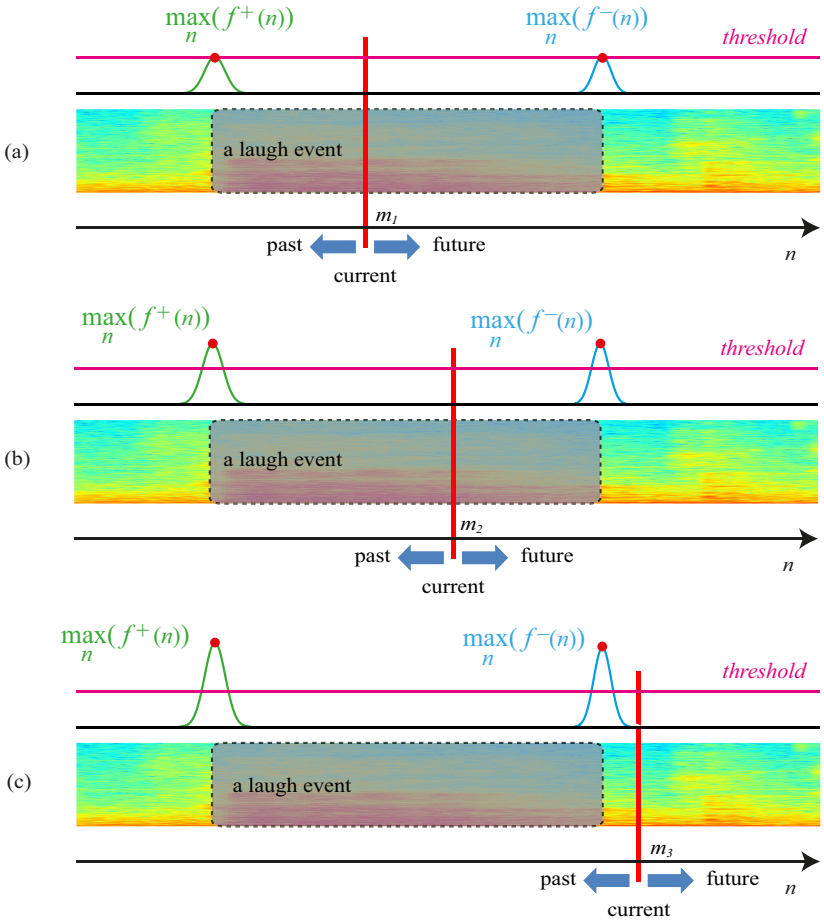

Figure 4.4: Illustration of early audio event detection in action. (a) The target event is detected when the onset and offset confidence score peaks reach the detection threshold. (b) The detected event remains locked and tracked during the event interval. (c) The target event is completed when the offset peak passes over the current time index.

4.1.4 Audio Event Detection in Action

The ability of early audio event detection requires realtime processing of an audio stream. It is very common that event instances of a certain target category occur more than once, one after another, in the stream. The detection system must be able to detect them sequentially. This section describes a proposed online detection mechanism which is illustrated in Figure 4.4. In the figure, the detector maintains two confidence scores, one for onset estimation and the other for offset estimation, centered at the current time index. The detector reads in audio segments from the data stream and keeps monitoring the occurrence of a target event. When an

audio segment arrives, the estimations obtained by this segment are accumulated to the confidence scores. As long as a pair of confidence score maxima above the predetermined detection threshold is found in chronological order (i.e the onset score peak in the past and the offset score peak in the future relative to the current time index), the target event is considered to be detected. Moreover, when the target event is detected, its temporal extent is determined as the interval starting from the estimated onset position and ending at the estimated offset position. Due to the monotonicity of the scoring function, during the event interval, both onset and offset scoring peaks remain above the threshold. This provides an automatic mechanism to track the target event. The detected event is locked and tracked as long as the order of its onset, the current time, and its offset remain unchanged. The event is considered to be completed when its offset passes the current time index. After that, the process is restarted to detect the upcoming target event. Therefore, at any time, the detector only needs to detect at most one target event.

4.2 Multi-channel Fusion

In this section, the framework proposed for multi-channel fusion will be described in detail, followed by the experiment on the ITC-Irst dataset [214] to demonstrate its efficiency.

4.2.1 The Additive Fusion Framework

The majority of works in literature have tackled single-channel AED mainly due to its simplicity. Very few attempts have considered taking advantage of multiple available distributed microphones to improve performance of the detection task. The distributed microphones offer more data with different views on the same problem. By integrating multiple audio channels to utilize spatial information of these microphones, performance improvements for the detection task can be expected. Unfortunately, so far, it is mostly not the case. It has been shown that a naive fusion strategy would deteriorate the system instead [204, 213, 215].

The multi-channel fusion framework proposed in this section is simple, yet efficient. In this framework, one regression forest detector presented in Chapter 3 is built for each individual channel. The fusion is then accomplished by summing up the estimation confidence scores of the per-channel detectors as demonstrated in Figure 4.5. In this simplified example, it is reasonable that an event can take

place at any location in the room and the power of the signals recorded by the microphones should be inversely proportional to their distances to the sound source. That is, those microphones closer to the source will most likely produce higher-SNR signals, allowing detection of the target event with higher confidence, and vice versa. The fusion framework acts as a sum over the microphones to collectively take advantage of the high-SNR signals and compensate the low-SNR ones.

Compared to the common multi-channel fusion approaches, i.e. the late fusion approach [76, 213, 215] which merges channel-wise decisions and the early fusion approach [108, 120] in which channel-wise features are fused, the proposed fusion scheme lies somewhere in between. Furthermore, while the common approaches cannot take advantage of intrinsic properties of the detection systems to guarantee performance gains, the proposed fusion scheme can secure performance improvement due to, again, the monotonicity of detection functions of the regression forest detectors. It will be shown in the experiments that monotonic gains of overall detection performance can be obtained as long as an additional data channel is added into the fusion system.

Let us denote the number of available channels as Q, and let the channels be indexed from 1 to Q. From Eqs. (3.26) and (3.27), the confidence scores $f_q^+(n)$ and $f_q^-(n)$ for event onset and offset estimations, respectively, at a time index n on a channel $q \in \{1, 2, \ldots, Q\}$ read

$$f_q^+(n) = \sum_m p_q^+(n \,|\, \mathbf{x}_{m,q}), \tag{4.6}$$

$$f_q^-(n) = \sum_m p_q^-(n \,|\, \mathbf{x}_{m,q}), \tag{4.7}$$

respectively. The fusion of the confidence scores of all Q channels can be done very naturally by accumulating the confidence scores from individual ones:

$$f_*^+(n) = \sum_{q=1}^{Q} \sum_m p_q^+(n \,|\, \mathbf{x}_{m,q}), \tag{4.8}$$

$$f_*^-(n) = \sum_{q=1}^{Q} \sum_m p_q^-(n \,|\, \mathbf{x}_{m,q}). \tag{4.9}$$

Here $f_*^+(n)$ and $f_*^-(n)$ denote the confidence scores of onset and offset estimation, respectively, after fusion. In addition, this fusion scheme preserves the monotonicity of the detection function since the final detection function is actually the sum of

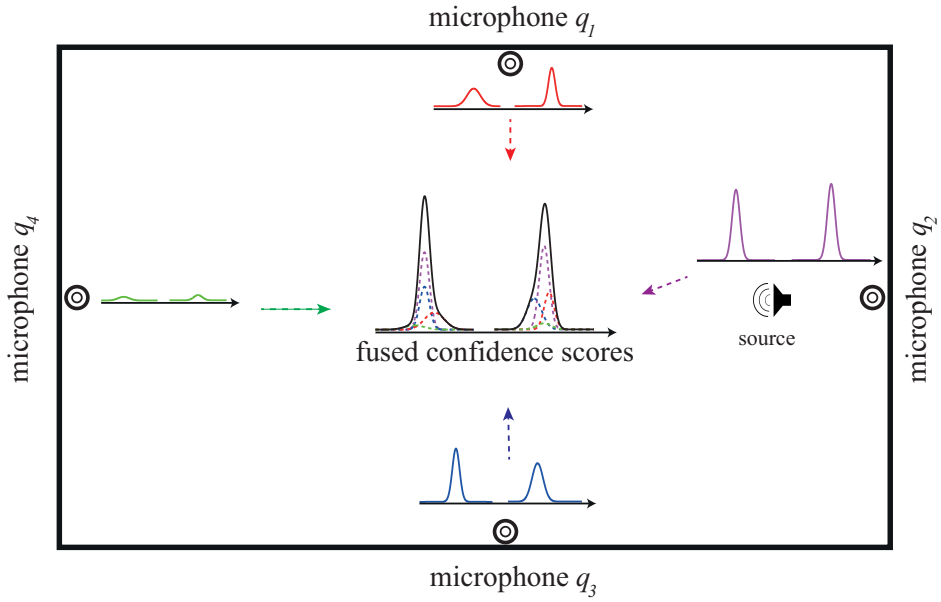

Figure 4.5: Illustration of the proposed multi-channel fusion scheme. Given four microphones q_1, q_2, q_3, and q_4 mounted at different positions, their estimation confidence scores for a target event occurring at the source are represented in different colors. The scores obtained by the microphone q_2 is expected to be largest since it is closest to the source whereas the furthest microphone q_4 should produce lowest scores. The fused confidence scores are collectively strengthened by the individual confidence scores, consolidating evidence about the target event occurrence.

individual monotonic detection functions. Therefore, the fusion system still has the capability of early event detection, similar to the channel-wise detectors.

It should be noticed that there will be temporal offsets among the channel-wise estimations due to different distances of distributed microphones to the sound source. However, when the microphones are close enough to each other, e.g. in order of several meters as in the ITC-Irst recording room shown in Figure 3.9, these offsets are negligible and can be safely ignored. In contrast, if the distances between microphones are large, synchronization may be necessary.

4.2.2 Experiments on Multi-Channel Fusion

This section presents the experiments conducted on the ITC-Irst dataset [214, 247] described in Section 3.3.1 to demonstrate the efficiency of the proposed multi-channel fusion framework. It would be redundant to use all 32 available microphones of this dataset in the experiments since many of them are located

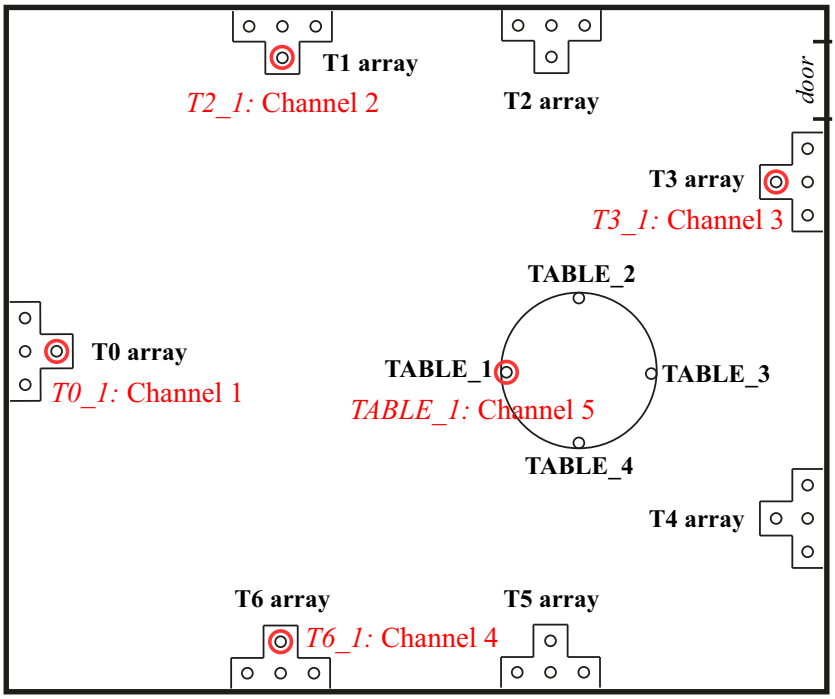

Figure 4.6: Five selected microphones of the ITC-Irst dataset [214], *T0_1*, *T1_1* , *T3_1*, *T6_1*, and *TABLE_1*, for the multi-channel fusion experiment.

very closely to each other. For example, those microphones in the same T-shaped arrays would not introduce much new spatial information into the fusion system. Instead, five following microphones were selected: *T0_1*, *T1_1* , *T3_1*, *T6_1*, and *TABLE_1* which are indexed as channel $\{1, 2, 3, 4, 5\}$ as shown in Figure 4.6. The first four of them are positioned at four side-walls while *TABLE_1* is positioned near the room center. Note that the microphone *TABLE_1* was used previously in the single-microphone experiments in Section 3.3.

To evaluate the channel-wise detection performance, the experiments in Section 3.3 were repeated for each of the selected channels, except for *TABLE_1* whose results are transferred from Section 3.3. Only the best weighting scheme, i.e. Weighting Scheme 3, was used in this study. The overall detection performances of the per-channel detectors as well as the fusion detector are shown in Table 4.1. It can be seen from the table that the locations of the microphones do influence the detection performances. Among the selected channels, Channel 4 is the most inferior, achieving 16.8% and 91.2% on ER and F1-score, respectively, while Channel 5 offers the best performance, obtaining 15.1% and 93.1% on ER and F1-score, respectively. These results can be explained by the fact that there were many

Table 4.1: Overall detection performance of the fusion system compared to the single-channel counterparts.

	Single channel					**Fusion**
	1	2	3	4	5	(1, 2, 3, 4, 5)
ER	15.3	15.1	16.3	16.8	15.1	12.3
F1-score	92.1	92.2	91.7	91.2	93.1	93.6

events taking place in the vicinity of *TABLE_1* during the recording sessions [214, 247].

The fusion system of all five selected channels lead to significant improvements on both ER and F1-score. Specifically, the obtained ER of the fusion system is lower than the average of the per-channel systems by 3.4% absolute while its F1-score is improved by 1.5% absolute on average. Compared to the best single-channel counterpart (Channel 5), the ER reduction and the F1-score improvement are 2.8% and 0.5% absolute, respectively. One may argue that the single Channel 5, whose microphone is located near the center of the room, can avoid computational overhead of multiple channel fusion with a slightly lower performance. However, in practice, depending on the spatial geometry of a specific application, it is not always possible to find a good compromise position for a single microphone. Furthermore, it is not possible to determine in advance which single channel is the best since the events can happen at any location within the room, not favoring a specific position of a microphone.

Figure 4.7 shows the variations of the overall ER and F1-score when the selected microphones are added into the fusion system one-by-one in the order of (1, 2, 3, 4, 5). It can be seen from the figure that, as long as a new channel is integrated into the system, the overall F1-score is further increased while the overall ER is further attenuated, except for the ER of the channel combination (1, 2) in Figure 4.7(a). The average ER reduction rate is about 0.8% while the average F1-score improvement rate is approximately 0.4%.

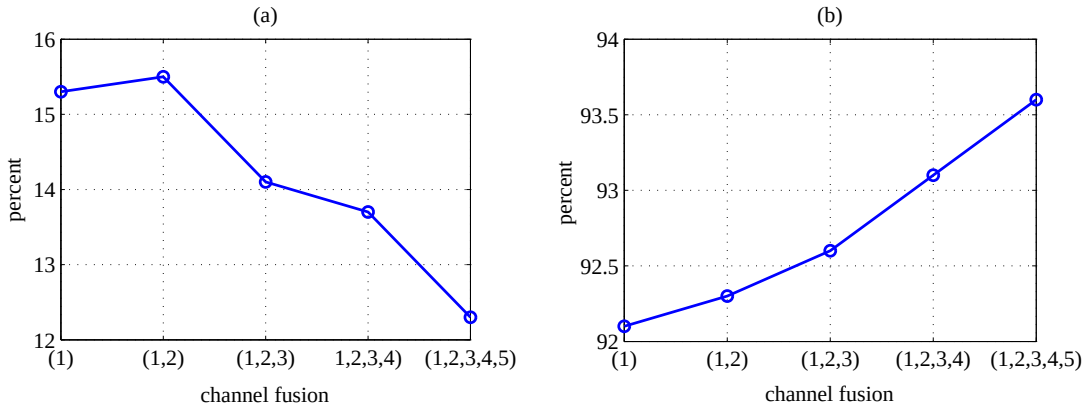

Figure 4.7: Variation of the overall detection performance when fusing the selected channels one-by-one in the order of (1, 2, 3, 4, 5) into the fusion system. (a) ER and (b) F1-score.

5 Audio Event Classification

This chapter will focus on audio event classification. As for a classification task in general, signal representation is particularly important to achieve good performance. Two learned representations are proposed in this chapter to take into account temporal structures of audio events: (1) audio phrases and (2) bank-of-regressors. The former is a generalized and improved version of the well-known BoW representation, where an audio phrase is defined as a sequence of multiple audio words. Using audio phrases enables capturing interaction between isolated audio words and, therefore, a certain degree of structural information. The objective of the bank-of-regressors representation is to combine the per-class regression forests in Chapter 3 into a bank for feature extraction. Since a regressor models the temporal structure of an event category, its response on an input event instance quantifies how well the event aligns to the temporal configuration of the category. The responses of those per-class regressors in the bank are used as structural features to represent the event instance.

5.1 Audio Phrases and Bag-of-Phrases Representation

The problem with the conventional BoW descriptors [7, 28, 160, 162, 180] is their reliance on unordered independent words. Hence, they are unable to take the structural information into account. In order to overcome this, audio phrases are proposed in this section to group audio words to encode the dependency between them and capture a certain degree of event temporal configuration. The idea is similar to the n-gram language models [161, 209] and the visual phrase concept in computer vision field [189, 218]. Afterwards, the bag-of-phrases (BoP) representation can be derived similarly to the BoW representation. However, this class of representations induces high dimensionality [161, 189, 218]. Specifically, the dimensionality of the BoP feature space grows exponentially with the codebook size. The curse of dimensionality hinders the conventional clustering-based codebook learning approaches which require a reasonably large number of audio words to

perform well. To alleviate this issue, a classifier which is discriminatively trained is alternatively used for codebook matching. The compact classifier-indexed discriminative codebook, in which the number of code words equals the number of target event categories, makes learning of higher-order audio phrases feasible.

5.1.1 Typical BoW Models

The BoW approach models an audio signal using its local features. Typically, the signal is decomposed into multiple segments, each of which is described by a vector of low-level features. The goal is, then, to quantize these local features using a codebook. The codebook can be built on the local features obtained from audio events in training data using a clustering method such as k-means [28] or GMM [85, 180]. In k-means based methods, a code word is represented by the cluster centroid. Within a probabilistic clustering framework of GMM, code words are represented by the Gaussian components. A local feature vector is then matched to a code word of the learned codebook with a certain matching weight. The weight assignment can be "hard" (e.g. with k-means) or "soft" (e.g. with GMM). The descriptor for an audio event is finally produced by simply accumulating the matching weights of different code words over all local features of the event.

5.1.2 Audio Phrases and Bag-of-Phrases Representation

While audio words in a BoW model are unordered, it is reasonable to group them into phrases which offer a higher semantic information level to enrich the representation. Towards this goal, the rationale behind this is to model the co-occurrences of the words in local neighborhoods, and therefore encode the temporal configurations of the events.

Suppose that one has learned a codebook $\mathcal{A} = \{a_1, a_2, \ldots, a_K\}$ of size K from the training data. Without loss of generality, let us denote an audio phrase $\mathcal{P}_{(a_{k_1}, a_{k_2}, \ldots, a_{k_N})}$ of order $N \geq 1$ as an ordered sequence of N code words $(a_{k_1}, a_{k_2}, \ldots, a_{k_N})$ where $a_{k_1}, a_{k_2}, \ldots, a_{k_N} \in \mathcal{A}$. As a result, there are totally K^N possible order-N audio phrases. The phrases reduce to the individual code words when $N = 1$.

Given the audio signal of a target event instance, it is firstly decomposed into a sequence of M segments $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M)$ where $\mathbf{x}_i$, for $1 \leq i \leq M$, denotes the low-level feature vector of the i-th segment. Each subsequence of N local segments

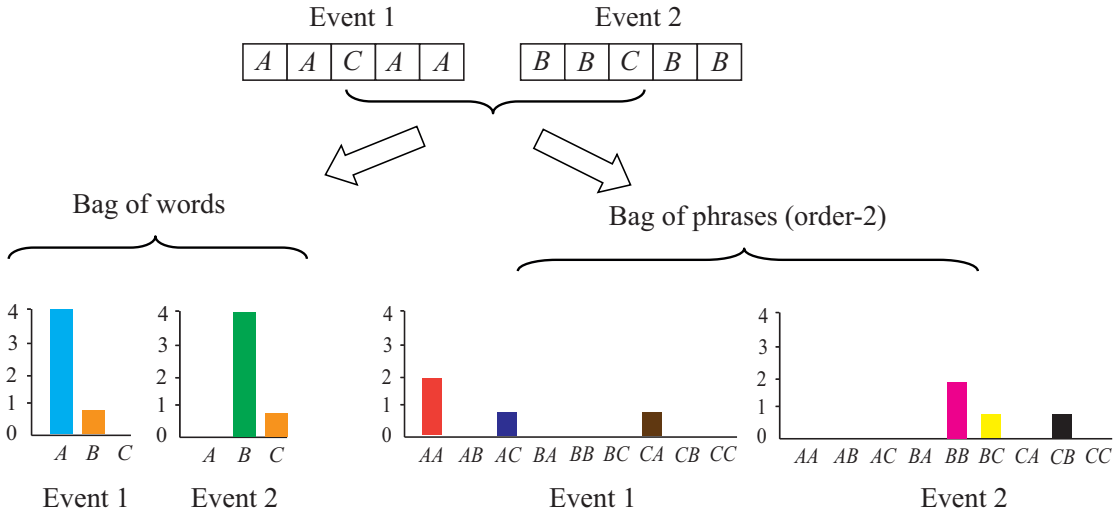

Figure 5.1: Illustration of the BoW and order-2 BoP descriptors produced for two simplified event instances. The events are simulated as two sequences of matched code words of the codebook $\mathcal{A} = \{A, B, C\}$.

$(\mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+N-1})$ is then matched to the order-N audio phrase $\mathcal{P}_{(a_{k_1}, a_{k_2}, \ldots, a_{k_N})}$ with the assigned weight given by

$$w\Big(\mathcal{P}_{(a_{k_1}, a_{k_2}, \ldots, a_{k_N})} \,|\, (\mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+N-1})\Big) = \prod_{j=1}^{N} w(a_{k_j} \,|\, \mathbf{x}_{i+j-1}). \tag{5.1}$$

Here, $w(a \,|\, \mathbf{x})$ is the assigned weight by matching the feature vector $\mathbf{x}$ to the code word $a \in \mathcal{A}$. In addition, $w(a \,|\, \mathbf{x})$ can be a likelihood function (e.g. using GMM-based clustering) or an indicator function (e.g. using k-means clustering). The accumulated weight by matching all possible order-N subsequences of the signal to the audio phrase $\mathcal{P}_{(a_{k_1}, a_{k_2}, \ldots, a_{k_N})}$ reads

$$w\Big(\mathcal{P}_{(a_{k_1}, a_{k_2}, \ldots, a_{k_N})} | (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M)\Big) = \sum_{i=1}^{M-N} w\Big(\mathcal{P}_{(a_{k_1}, a_{k_2}, \ldots, a_{k_N})} | (\mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+N-1})\Big). \tag{5.2}$$

Eventually, the target event instance is represented by the weights obtained by matching it to all possible order-N audio phrases, i.e. the order-N BoP representation.

Figure 5.1 illustrates the BoW and BoP representations for two simplified examples. The BoP representation exhibits a denoising property. Usually, if there exist shared features between audio events in which two events may have similar

local features, they should occur in patterns of multiple consecutive segments to be useful for the classification task. The intermittent occurrence of a code word, which is significantly different from its neighbors, should be considered as noise, and, therefore, should be filtered out. As shown in the example in Figure 5.1, two different events have the code word "C" in common which should be considered as noise. Comparison of the BoW descriptors, e.g. using histogram intersection distance, will result in a positive similarity value due to the positive weights assigned to "C", whereas, the similarity value is zero when using the BoP descriptors. In other words, using the BoP descriptors has canceled out the noisy "C" and increased the distinction between two events.

5.1.3 Learning Discriminative and Compact Codebooks

The performance of conventional BoW models heavily depends on the codebook size. More often than not, the codebook size is multiple-order larger than the number of target event categories. To support this argument, Figure 5.2 illustrates the performances of different BoW models on four datasets: ITC-Irst [214], UPC-TALP [24], Freiburg-106 [205], and NAR [139], varying as a function of the codebook size. The codebooks were constructed using k-means clustering and the SVM classifiers' performances with respective to the best kernels (among linear, RBF, χ^2, and histogram intersection kernels) were used for this plot (more details in Section 5.3.3). It can be seen that a codebook size of 200 is a reasonable choice for the Freiburg-106 dataset, for example. Given the fact that the number of target event categories of this dataset is 22, the codebook size is about ten times larger. On the other hand, using this codebook, the feature space induced by the order-N BoP has the dimensionality of 200^N. It is $4{\times}10^4$ with $N = 2$ and $8{\times}10^6$ with $N = 3$. This exponential growth of dimensionality makes clustering-based codebook learning inappropriate for the BoP models.

A method to learn a compact codebook in a supervised manner is proposed here to alleviate the high-dimensionality problem. While the conventional clustering methods ignore the labeling information, integrating them into the codebook construction offers more discrimination power [147]. Inspired by this, rather than clustering, a segment-wise classification model is employed for codebook matching. As a result, the codebook size is equal to the number of target event categories, and the dimensionality of the BoP descriptors will be drastically reduced. Although multiple one-vs-rest binary classifiers would suite this goal, random-forest

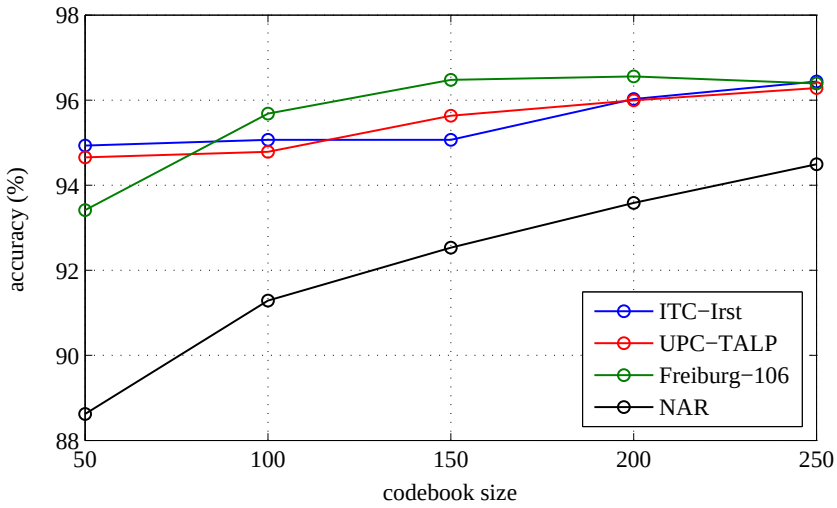

Figure 5.2: Classification accuracy of the BoW models as a function of codebook size.

classification [18] is used here to learn a multi-class classifier at once. Moreover, random-forest classification supports probability output. It turns out that both hard and soft codebook matching can be investigated simultaneously.

Suppose that there are C event categories of interest, and hence, the number of code words K equals C. Furthermore, suppose that the segment-wise random-forest classifier $\mathcal{M}_{\text{ev}}$ has been trained using the training audio segments. The soft assigned weight by matching an unseen audio segment represented by the feature vector $\mathbf{x}$ to a code word (also identical to the class label) $c \in \{1, 2, \ldots, C\}$ reads

$$w(c \,|\, \mathbf{x}) = P(c \,|\, \mathbf{x}). \tag{5.3}$$

Here, $P(c \,|\, \mathbf{x})$ is the posterior probability that $\mathbf{x}$ is classified as the class c by the classifier $\mathcal{M}_{\text{ev}}$. On the other extreme, the hard assignment yields the weight

$$w(c \,|\, \mathbf{x}) = \mathbb{I}(\hat{c} = c \,|\, \mathbf{x}), \tag{5.4}$$

where

$$\hat{c} = \underset{c \in \{1,2,\ldots,C\}}{\arg\max} \; P(c \,|\, \mathbf{x}), \tag{5.5}$$

with the indicator function $\mathbb{I}(\cdot)$ given in Eq. (3.34).

It will be shown in the experiments that the hard assignment scheme produces much sparser descriptors compared to those obtained with the soft assignment scheme at the cost of slightly lower recognition accuracies.

5.2 Bank-of-Regressors Representation

This proposed representation utilizes the category-specific random regression forests presented in Section 3.1 to extract structural features for audio events. Given an unseen event instance, via a learned regression model, its local features are used to estimate the boundaries of the event. By doing this, the "shape" of the audio event, i.e. its temporal extent, has been implicitly modeled as a constellation of its local features [205]. Since the regressor models the relative positions of the audio segments to the event onsets and offsets, their predicted confidence scores in Eqs. (3.26) and (3.27) can be reasonably considered as structural measures. The outputs by evaluating the regression forest on the audio event quantify how well the event aligns to the temporal configuration of the event category modeled by the forest.

5.2.1 Regressors for Structural Measurements

Suppose that a regressor $\mathcal{F}_c$ has been learned for a target event class c as in Section 3.1. Given the audio signal of an event instance, it is firstly decomposed into a sequence of M segments $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M)$. Being presented with an audio segment $\mathbf{x}_m$ at the time index m, where $1 \leq m \leq M$, as input, the regressor produces estimates for the event onset and offset positions in terms of the probability density functions $p^+(n \,|\, \mathbf{x}_m, c)$ and $p^-(n \,|\, \mathbf{x}_m, c)$ given in Eqs. (3.24) and (3.25), respectively. From Eqs. (3.30) and (3.31), the respective onset and offset estimation confidence scores $f^+_{c,\mathbf{x}_m}(n)$ and $f^-_{c,\mathbf{x}_m}(n)$ are computed by

$$\begin{aligned} f^+_{c,\mathbf{x}_m}(n) &= p^+(n, c \,|\, \mathbf{x}_m) \\ &= \boldsymbol{\lambda}(c \,|\, \mathbf{x}_m)\, p^+(n \,|\, \mathbf{x}_m, c) \\ &= P(c \,|\, \mathbf{x}_m)\, p^+(n \,|\, \mathbf{x}_m, c), \qquad (5.6) \\ f^-_{c,\mathbf{x}_m}(n) &= p^-(n, c \,|\, \mathbf{x}_m) \\ &= \boldsymbol{\lambda}(c \,|\, \mathbf{x}_m)\, p^-(n \,|\, \mathbf{x}_m, c) \\ &= P(c \,|\, \mathbf{x}_m)\, p^-(n \,|\, \mathbf{x}_m, c). \qquad (5.7) \end{aligned}$$

In Eqs. (5.6) and (5.7), Weighting Scheme 3 given in Eq. (3.36) is specifically used where $P(c\,|\,\mathbf{x}_m)$ is the posterior probability that the local feature $\mathbf{x}_m$ is matched to the event class c. The estimates by all audio segments read

$$f_c^+(n) = \sum_{m=1}^{M} f_{c,\mathbf{x}_m}^+(n), \tag{5.8}$$

$$f_c^-(n) = \sum_{m=1}^{M} f_{c,\mathbf{x}_m}^-(n). \tag{5.9}$$

Since it is intended to estimate the onset and offset positions separately, the regression confidence scores $f_c^+(n)$ and $f_c^-(n)$ can be interpreted as the measures for forward and backward structures of the event, respectively. Finally, the onset and offset confidence score maxima are averaged to produce the overall structural alignment ϕ_c of the event class c measured on the input audio event:

$$\phi_c = \frac{1}{2}\left(\max_n\left(f_c^+(n)\right) + \max_n\left(f_c^-(n)\right)\right). \tag{5.10}$$

The value of ϕ_c can be interpreted as how much the input event instance aligns to the temporal configuration of the target event category c modeled by the corresponding regressor $\mathcal{F}_c$.

5.2.2 Bank-of-Regressors Representation

In fact, the accumulated confidence scores given in Eqs. (5.8) and (5.9) can be directly adapted for the classification task, for example, with a winner-take-all voting scheme. However, this will ignore the shared features between different classes which are important to boost the recognition performance [185]. To resolve this issue, the category-specific regressors can be stacked in a bank as in Figure 5.3 for feature extraction. The regressor bank then plays the role of a mid-level feature extractor to produce an intermediate representation for the audio event. The responses of the regressors on a target event quantify the alignment of the event to the structures of different event classes, and hence, encode their shared features. Formally, the sequence of audio segments $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M)$ of the input signal has been transformed into a compact bank-of-regressors (BoR) descriptor $\boldsymbol{\phi} = \begin{pmatrix} \phi_1 & \phi_2 & \ldots & \phi_C \end{pmatrix}^{\mathsf{T}} \in \mathbb{R}_+^C$ where ϕ_c, for $1 \leq c \leq C$, is given in Eq. (5.10). As a result, the audio event is embedded in the space spanned by the responses of

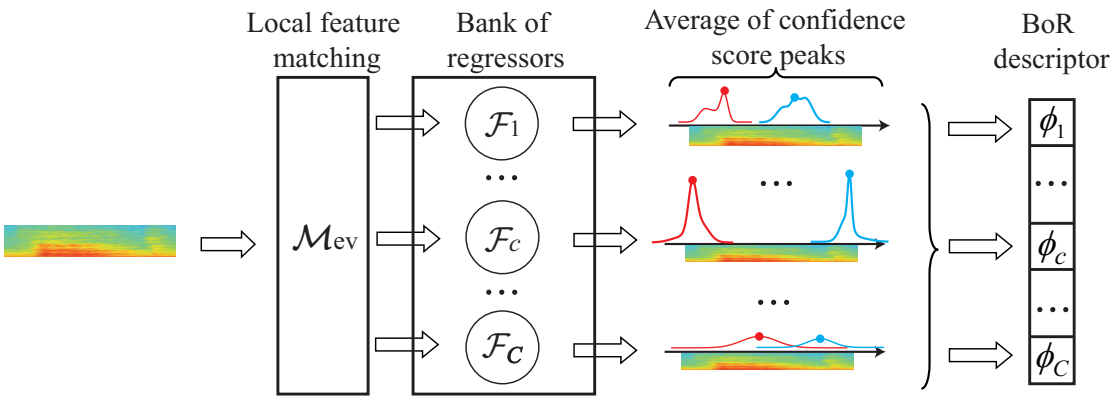

Figure 5.3: Extraction of the BoR descriptor. The local feature matching is performed by the segment-wise classifier $\mathcal{M}_{\text{ev}}$. The class-specific regressors $\mathcal{F}_c$ produce confidence scores for onset and offset positions. The confidence score maxima are then averaged to yield structure alignment measurements ϕ_c of the BoR descriptor $\boldsymbol{\phi}$.

the regressors. Each entry ϕ_c is then divided by the maximum value of ϕ_c in the training events for normalization. The vector $\boldsymbol{\phi}$ is finally normalized by ℓ_1-norm.

Besides the temporal coding and shared feature encoding ability, the BoR descriptor is compact, meaning that its dimensionality is equal to the number of target event categories. Last but not least, since the BoR descriptors are semantically rich representations, even simple linear classification models trained on them are able to obtain good classification accuracy.

For illustration purposes, Figure 5.4 shows the normalized responses of the regressor bank on typical examples of different categories of the ITC-Irst dataset. Note that the events are zero-padded at the beginning and the end to make them five times longer before regression to account for event duration variations. It can be observed that some examples (e.g. “paper warping” and “applause”) are very discriminative, for which a winner-take-all voting scheme should be adequate for recognition. However, such a voting scheme would yield wrong recognition on many other classes (e.g. “door slam” and “key jingle”). A linear combination of the responses to incorporate the shared features can overcome this in the BoR descriptor.

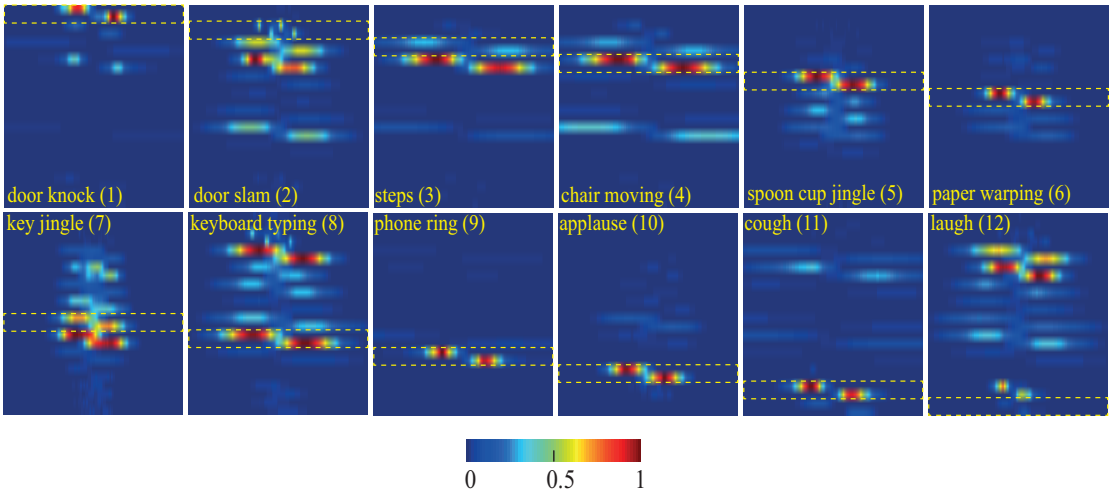

Figure 5.4: Responses of regressor bank on audio events of different classes of the ITC-Irst dataset. The numbers in brackets indicate the class identity. For an event of class c, the responses of the regressor $\mathcal{F}_c$ are located in the dash-line boxes, the onset score on one row followed by the offset score on the other row.

5.2.3 Combination with Unstructured Features

The BoR descriptor is expected to work well for event categories which expose strong temporal structures. On the contrary, for weakly structured events, such as those with impulse-like signals (e.g. "door slam"), unstructured features (e.g. BoW) would be more useful. Therefore, it is reasonable to combine both types of descriptors to exploit both of their strength. Although improvements are experimentally seen when combining the BoR descriptor and the standard BoW descriptor, it is quite costly to build an additional BoW system for this fusion. Alternatively, the random-forest classifier $\mathcal{M}_{\text{ev}}$ can be utilized to form an unstructured descriptor with very little induced cost.

Given the sequence of audio segments $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M)$ of an event instance, the unstructured descriptor $\boldsymbol{\vartheta} = \begin{pmatrix} \vartheta_1 & \vartheta_2 & \ldots & \vartheta_C \end{pmatrix}^{\mathsf{T}} \in \mathbb{R}_+^C$ can be obtained where

$$\vartheta_c = \frac{1}{M} \sum_{m=1}^{M} P(c \,|\, \mathbf{x}_m) \tag{5.11}$$

for $1 \leq c \leq C$. The vector $\boldsymbol{\vartheta}$ is then further normalized by ℓ_1-norm.

Different descriptors are combined using the extended Gaussian kernel [241]:

$$\kappa(e_i, e_j) = \exp\left(-\sum_{k\in\{\boldsymbol{\phi},\boldsymbol{\vartheta}\}} \frac{1}{\bar{D}^k} D_{\chi^2}\left(e_i^k, e_j^k\right)\right), \tag{5.12}$$

where $D_{\chi^2}\left(e_i^k, e_j^k\right)$ denotes the χ^2 distance between the audio events e_i and e_j with respect to the feature channel k. $\bar{D}^k$ is the mean value of the χ^2 distances between the training samples for the channel k. Finally, an SVM classifier with the kernel $\kappa(\cdot,\cdot)$ defined in Eq. (5.12) is trained for classification.

It is worth noting that the classifier $\mathcal{M}_{\text{ev}}$ in this section is identical to the one used as the codebook matcher for audio phrases in Section 5.1.3. Therefore, the unstructured descriptor $\boldsymbol{\vartheta}$ is equivalent to the BoW model with the soft codebook matching scheme in Section 5.1.3. In addition, the components in Figure 5.3 (i.e. the segment-wise classifier $\mathcal{M}_{\text{ev}}$ and the class-specific regression $\mathcal{F}_c$) are parts of the detection pipeline in Figure 3.6. That is, no additional components need to be built to extract the learned representations presented in this chapter (i.e. the BoP and BoR representations) but using the same ingredients of the detection pipeline in Figure 3.6. This turns out to be a great benefit when the classifiers trained on these in situ features are used to verify the detected events of the detection system for false positive reduction in Chapter 7.

5.3 Experiments

This section describes the experiments conducted on different datasets to analyze performances of the presented audio event representations for the classification task. The performances obtained by these representations are also compared with those of different baselines to demonstrate their efficiency.

5.3.1 Datasets

In addition to the ITC-Irst dataset [214, 247] used in Section 3.3.1, three other datasets were employed for the experiments, including UPC-TALP [24], Freiburg-106 [205], and NAR [139]. These datasets were recorded in different environments, and hence, have dissimilar reverberation characteristics. Moreover, they also differ in the complexity since the number of target event categories varies from one to another. The datasets are summarized in Tables 5.1, 5.2, and 5.3.

- **UPC-TALP dataset [24].** Similar to the ITC-Irst dataset [214, 247], this dataset was recorded in a meeting-room environment. It is multi-channel and multimodal (i.e. audio and video) and contains recording sessions of both isolated and spontaneous audio events. However, for the classification task, only recordings with isolated events on a single audio channel (channel 10 [24]) were used in the experiments. There were eight recording sessions where six different participants performed ten times each event. Totally, there are 1,418 instances of eleven event categories. Following the setting in [151], leave-one-session-out cross-validation was conducted. At each time, seven sessions were used for training and the remaining one was used for testing. The average accuracy is finally reported.

- **Freiburg-106 dataset [205]**. This dataset was collected using a consumer-level dynamic cardioid microphone in kitchen and bathroom environments. It consists of 1,476 audio-based human activities of 22 categories. Particularly, several sources of ambient noise (e.g. PC fans whirring [205]) were also presented during the recording process. As in [205], the dataset was split into training and test sets as in [205][1]. For each category, every second example was included into the test set and the remaining ones were moved into the training set.

- **NAR dataset [139]**. This dataset was recorded using the frontal bandpass microphone of a NAO robot in both home and office environments. The recording process suffered from interference of robot-head fan noise. Besides nonspeech events, there exist several speech word categories, however, they were treated as audio events in general. Overall, the dataset consists of 852 examples of 42 event categories, each of which has 20 or 21 event instances. As in [139], the dataset was randomly divided into ten equal folds and leave-one-fold-out cross-validation was conducted. Finally, the average cross-validation performance accuracy is reported.

5.3.2 Parameters

The audio signals were firstly downsampled to 16 kHz. Each audio event instance was decomposed into a sequence of 50 ms segments with a Hamming window and

[1]This is based on unofficial communication with the authors of [205].

Table 5.1: Summary of the UPC-TALP dataset [24]. Eight recording sessions of the dataset are denoted as {S1, S2, ..., S8}.

Event Type	**# event instance**								
	S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**Total**
knock (door, table)	9	8	10	10	10	8	11	13	79
door slam	17	15	19	20	40	37	56	52	256
steps	10	10	8	23	43	34	28	50	206
chair moving	19	37	32	22	23	38	34	40	245
spoon (cup jingle)	10	11	13	11	10	15	11	15	96
paper work	9	11	10	8	17	12	12	12	91
key jingle	11	11	11	8	0	13	10	18	82
keyboard typing	10	10	13	12	10	13	10	11	89
phone ringing	11	18	11	14	8	11	13	15	101
applause	9	5	9	11	12	9	14	14	83
cough	10	10	12	13	9	13	11	12	90
Total	125	146	148	152	182	203	210	252	1,418

Table 5.2: Summary of the Freiburg-106 dataset [205].

Event Type	**# event instance**	**Event Type**	**# event instance**
background	47	microwave	92
food bag opening	80	microwave bell	24
blender	60	microwave door	86
cornflakes bowl	36	plates sorting	135
cornflakes eating	43	stirring cup	59
pouring cup	22	toilet flush	124
dish washer	89	tooth brushing	29
electric razor	83	vacuum cleaner	79
flatware sorting	40	washing machine	67
food processor	35	water boiler	65
hair dryer	66	water tap	115
Total	1,476		

Table 5.3: Summary of the NAR dataset [139].

Event Type	**# event instance**	**Event Type**	**# event instance**
eating	21	tongue clic	20
choking	21	one	20
cutlery	21	two	20
fill a glass	21	three	20
running tape	21	four	20
open/close a drawer	21	five	20
move a chair	21	six	20
open microwave	21	seven	20
close microwave	21	eight	20
microwave (alarm)	21	nine	20
fridge (alarm)	21	ten	20
toaster (alarm)	21	hello	20
door close	20	left	20
door open	20	right	20
door key	20	turn	20
door knock	20	move	20
ripped paper	20	stop	20
zip	20	Nao	20
(another) zip	20	yes	20
fingerclap	20	no	20
handclap	20	what	20
Total	852		

a step size of 10 ms. An analysis of various segment sizes $\{30, 40, \ldots, 100 \text{ ms}\}$ was performed and will be reported in Section 5.3.5 to show how the classification performance changes with them. The set of low-level features in Section 3.3.3 was extracted to represent an audio segment.

The segment-wise classifier $\mathcal{M}_{\text{ev}}$ was trained with random-forest classification [18] with 200 trees. For the purpose of classification, an audio segment was labeled by the label of the event from which it stemmed. The class-specific regressors $\mathcal{F}_c$ were learned with the same settings as in Section 3.3.2. To extract the BoP and BoR descriptors for the training examples, ten-fold cross-validation was accomplished on the training data. Note that, as in Section 3.3, it is sufficient to perform cross-validation for the classifier $\mathcal{M}_{\text{ev}}$. The regression forests $\mathcal{F}_c$ trained with the whole training data were employed for both testing and cross-validation purposes.

The final event classification models were trained with one-vs-one SVMs with four different kernels, including linear, χ^2, histogram intersection (hist. for short), and RBF. The first three kernels are given by

$$\kappa(\mathbf{x}, \mathbf{z}) = \mathbf{x}^{\mathsf{T}}\mathbf{z}, \tag{5.13}$$

$$\kappa(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{D} \frac{2x_i z_i}{x_i + z_i}, \tag{5.14}$$

$$\kappa(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{D} \min(x_i, z_i). \tag{5.15}$$

The RBF kernel is given in Eq. (3.49).

In the above equations, $\mathbf{x}$ and $\mathbf{z}$ denote two D-dimensional input feature vectors. Grid search was performed with 10-fold cross-validation to search for the hyperparameters of the SVMs. For the hyperparameter C that trades off errors of the SVMs on training data and margin maximization, a coarse grid search in the set $\left\{2^k; k \in \{-2, -1, \ldots, 8\}\right\}$ was conducted first. It was then followed by a fine search in the set $\left\{2^k; k \in \{C_* - 1, C_* - 0.75, \ldots, C_* + 1\}\right\}$ where C_* denotes the optimal value found by the coarse search. A similar grid search was also conducted for the γ parameter of the RBF kernel given in Eq. (3.49). The *libSVM* library [31] was used for training and testing.

5.3.3 Baseline Systems

The three following baseline systems were developed for performance comparison.

- **Bag-of-words model (BoW).** BoWs have been widely used for audio event classification [7, 28, 85, 160, 162, 180]. A typical BoW model was implemented here and is used as a baseline system. Using this model, an audio event, which is decomposed into a set of segments, is represented by a histogram of code words. The k-means clustering algorithm [112] was used for codebook learning. The code words were represented as the cluster centroids, and codebook matching was based on the Euclidean distance. Since the performance of the BoW model heavily depends on the codebook size, the codebook size was varied to have values in the set of $\{50, 100, \ldots, 250\}$. The classifiers were also trained using SVMs with linear, χ^2, hist., and RBF kernels. Out of various settings (i.e. different codebook sizes and kernels), the one with the best performance was retained for comparison.

- **Pyramid bag-of-words model (PBoW).** The temporal structure of audio events has been shown important for the recognition task [35, 51, 116, 161, 180]. However, the BoW baseline, which considers an event instance as a set of unordered audio segments, is unable to take into account the structural information. As an improvement of the BoW, PBoW descriptors introduce a certain degree of temporal structure of audio events by extracting and combining BoW descriptors on different pyramid levels. This technique was first described in the seminal work by Lazebnik et al. [123] for natural scene classification in the field of computer vision. It has recently been shown to be useful for audio event representation and reported state-of-the-art performance on different benchmark datasets [180]. In addition to different codebook sizes as in the BoW baseline, $\{2, 3, 4\}$ pyramid levels were exploited here. Again, the classifiers were trained using SVMs with linear, χ^2, hist., and RBF kernels as in the proposed systems. The system which yields the best performance will be retained for comparison.

- **Bank-of-regressors max voting (BoR-MV).** This baseline directly uses the confidence scores outputted by the individual regressors in the bank (cf. Figure 5.3) for classification with a winner-take-all voting scheme. An unseen event instance will be assigned the class label that corresponds to the regressor producing the best structural alignment feature:

$$\hat{c} = \underset{c \in \{1,2,\ldots,C\}}{\arg\max}\ \phi_c, \tag{5.16}$$

Table 5.4: The best classification accuracies (%) obtained by the **BoW** and **PBoW** baselines. The settings (i.e. the SVM kernel and the codebook size) corresponding to the obtained results are also shown in brackets.

	BoW	**PBoW-2**	**PBoW-3**	**PBoW-4**
ITC-Irst	96.4 (χ^2, 250)	**96.7** (χ^2, 250)	95.9 (χ^2, 250)	94.0 (χ^2, 200)
UPC-TALP	96.3 (hist., 250)	96.5 (hist., 250)	**96.6** (χ^2, 250)	96.2 (χ^2, 200)
Freiburg-106	**96.6** (hist., 200)	**96.6** (hist., 200)	96.3 (hist., 200)	95.8 (hist., 250)
NAR	94.5 (χ^2, 250)	95.8 (χ^2, 200)	**96.4** (χ^2, 250)	96.1 (χ^2, 250)

where ϕ_c is given in Eq. (5.10) and $\hat{c}$ denotes the predicted label. The purpose of this baseline is to demonstrate the importance of integrating the responses of the class-specific regressors into the BoR feature vector to encode the shared features between different classes.

As for the proposed systems, the experiments on the baselines were repeated five times and their average performances are reported.

5.3.4 Experimental Results

The performance obtained by the baseline systems and the proposed systems on the experimental datasets will be presented and analyzed in this section, followed by their performance comparison.

5.3.4.1 Performances of BoW and PBoW baselines

Regarding the **BoW** and **PBoW** baseline systems, their best performances with respect to different settings are shown in Table 5.4. For clarity, let us denote the **PBoW** system with l pyramid levels as **PBoW-**l where $l \in \{2, 3, 4\}$. For all these baselines, the χ^2 and hist. kernels are the best ones which are expected for histogram-based representations. Furthermore, the codebook size exhibits an important role. Figure 5.5 shows the baseline performances for the best kernel varying as function of the codebook size. As can be seen from this figure, the

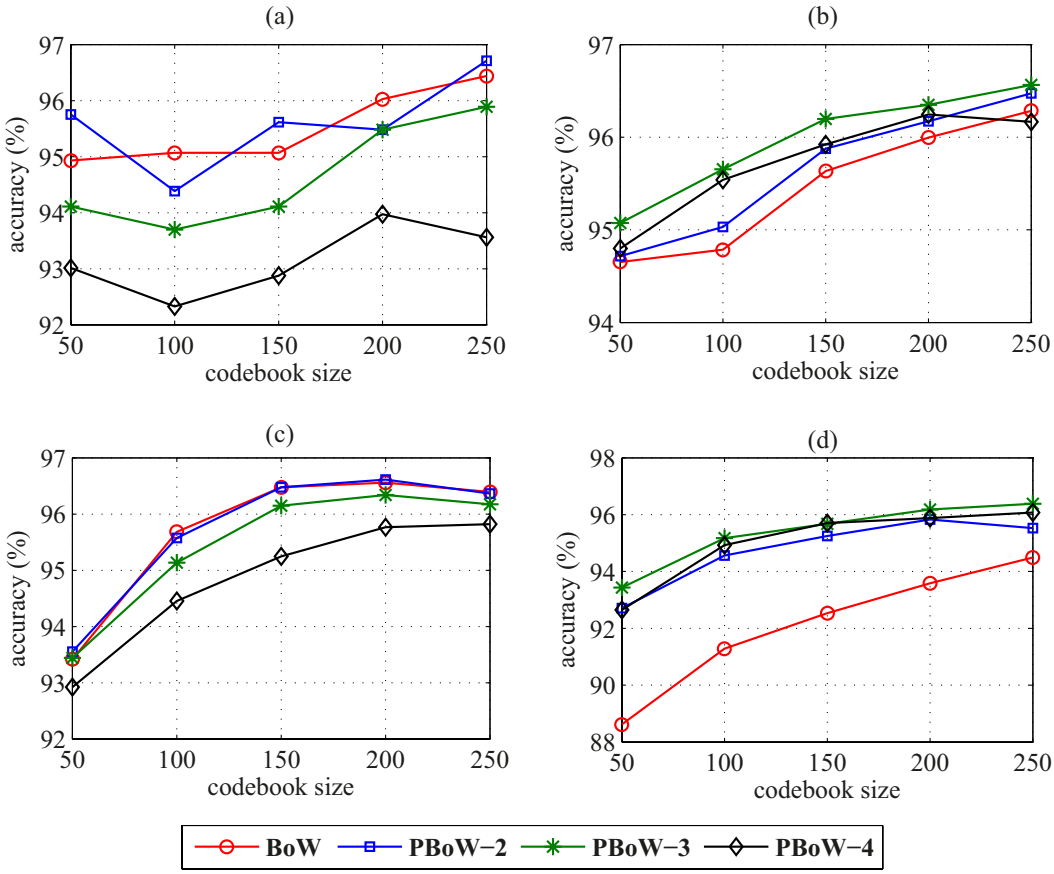

Figure 5.5: The best classification accuracy obtained by the **BoW** and **PBoW** baselines for different codebook sizes. (a) ITC-Irst, (b) UPC-TALP, (c) Freiburg-106, and (d) NAR.

codebook size needs to be large enough to guarantee a good performance. Over all studied datasets, the best performances are achieved with the codebook sizes of 200 and 250, respectively.

For comparison, in general, the **PBoW** baselines perform better than the **BoW** ones although the influence of the pyramid level varies for different datasets. Specifically, pyramid level $l = 2$ is found to be the best for ITC-Irst and Freiburg-106 whereas UPC-TALP and NAR are best classified with pyramid level $l = 3$. Overall, going to higher pyramid levels impairs the **PBoW** performance due to too fine-grained representation of the input signals.

5.3.4.2 Performances of BoP systems

Let us denote an order-N **BoP** system as **BoP-*N***. The influence of different orders $N = \{1, 2, 3\}$ on both hard and soft codebook matching will be examined in this section. Their classification accuracies are shown in Table 5.5.

Overall, among four kernels, χ^2 and hist. kernels are the best since the BoP itself is a histogram-like representation. This is indeed computationally convenient since these additive kernels are fast to evaluate [137]. There is only one exception for the soft **BoP-2** case on the ITC-Irst dataset for which the best result is obtained with the RBF kernel. However, the absolute accuracy gap between this result and that obtained with the χ^2 kernel is only 0.3%, which is marginal. Also, the performances of the simple linear classifiers are also comparable with those of the nonlinear ones in several cases. For instance, their accuracies on the Freiburg-106 dataset using hard **BoP-1**, hard **BoP-2**, and hard **BoP-3**, are just 0.3%, 0.1%, and 0.8% lower than the highest accuracies, respectively.

Between the hard and soft **BoP**s, the latter perform better than the former over all the experimental datasets and the phrase orders. Specifically, Table 5.6 shows the absolute accuracy gains of the soft **BoP** systems over the hard **BoP** counterparts with respect to different datasets and kernels. The performance gap is most likely due to the fact that large quantization errors have been introduced into the hard **BoP**s by using the discrete class label frequencies. In contrast, such quantization errors are mitigated in the soft **BoP**s. However, the accuracy gains are achieved at the cost of increasing percentages of zero entries in the soft **BoP** descriptors as shown in Table 5.7. As can be seen from the table, the hard **BoP**s largely contain zero entries and their proportion exponentially grows with the phrase order N. Conversely, the numbers of the zero entries are minor in the case of the soft **BoP**s and their growth rate is significantly slower. Therefore, the computation and storage for the hard **BoP**s can be much more efficient than those for the soft ones. In general, from the soft to hard **BoP**s, a wide range of **BoP** representations with different degrees of quantization can be derived.

When increasing the phrase order N, a higher-level dependency between isolated words can be encoded and, as a result, a better classification accuracy can be expected. However, it is also expected that the performance will level off at a certain point when the high-order dependency is not generalized for both training and testing data. As can be seen from Table 5.5, for the soft **BoP**s absolute accuracy gains of 0.7% and 0.8% are achieved on the UPC-TALP with the χ^2 and hist.

Table 5.5: Classification accuracy of the **BoP** systems.

	hard **BoP-1**				hard **BoP-2**				hard **BoP-3**			
	linear	χ^2	hist.	RBF	linear	χ^2	hist.	RBF	linear	χ^2	hist.	RBF
ITC-Irst	95.1	**96.8**	95.3	95.8	95.8	**96.8**	96.0	95.8	95.6	96.2	95.3	95.6
UPC-TALP	94.7	95.4	95.2	95.3	94.5	95.8	95.3	94.7	94.7	**96.0**	95.5	94.8
Freiburg-106	97.5	97.7	97.8	97.4	97.6	97.7	97.6	97.6	97.1	97.8	**97.9**	97.1
NAR	94.9	**95.3**	**95.3**	95.0	93.8	94.5	94.7	93.5	92.1	93.1	93.3	91.6
	soft **BoP-1**				soft **BoP-2**				soft **BoP-3**			
	linear	χ^2	hist.	RBF	linear	χ^2	hist.	RBF	linear	χ^2	hist.	RBF
ITC-Irst	95.8	97.0	95.8	96.2	96.8	97.0	96.8	**97.3**	96.8	96.8	97.1	96.0
UPC-TALP	95.8	96.2	96.2	95.8	95.9	96.9	97.0	96.3	96.1	**97.2**	**97.2**	96.8
Freiburg-106	98.6	98.4	98.3	98.5	97.7	98.7	**98.9**	97.7	97.2	98.7	**98.9**	97.0
NAR	96.9	97.0	97.1	96.9	96.7	**97.6**	**97.6**	96.5	92.6	**97.6**	**97.6**	95.2

Table 5.6: Absolute classification accuracy gains of the soft **BoP** systems against the hard **BoP** counterparts. The gains are averaged over different orders $N = \{1, 2, 3\}$.

	linear	χ^2	hist.	RBF
ITC-Irst	1.0	0.3	1.0	0.8
UPC-TALP	1.3	1.0	1.5	1.4
Freiburg-106	0.4	0.9	0.9	0.4
NAR	1.8	3.1	3.0	2.8

Table 5.7: Percentage of the number of zero entries in the **BoP** descriptors.

		BoP-1	**BoP-2**	**BoP-3**
ITC-Irst	hard	43.8	84.3	97.4
	soft	0.0	0.0	0.1
UPC-TALP	hard	58.0	89.0	98.1
	soft	0.9	3.6	8.3
Freiburg-106	hard	81.5	97.8	99.8
	soft	8.5	23.1	38.6
NAR	hard	83.3	99.2	99.9
	soft	6.6	21.7	40.0

kernels, respectively, when N increases from 1 to 2. Accordingly, further absolute accuracy improvements of 0.3% and 0.2% are obtained with **BoP-3** compared to **BoP-2**. However, on the Freiburg-106 and NAR datasets, no additional gains are seen for both χ^2 and hist. kernels when moving from **BoP-2** to **BoP-3**. It is even worse in case of the ITC-Irst, the **BoP-3**'s accuracy is 0.2% lower absolute than that of **BoP-2**. Unfortunately, these patterns are not seen for the hard **BoP**s. Obviously, the quantization errors are amplified, leading to performance drops with increasing phrase orders.

Last but not least, compared to the nondiscriminative codebook (i.e. the **BoW** baselines), the discriminative codebook (i.e. the **BoP-1**) offers much more favorable results. With the χ^2 kernel, the hard **BoP-1** systems bring up absolute accuracy improvements of 0.4%, 1.1%, and 0.8% compared to the best **BoW** baselines on the

Table 5.8: Classification accuracy of the BoR-based systems.

	BoR-MV	**BoR**				**BoR+**
		linear	χ^2	hist.	RBF	
ITC-Irst	95.5	98.1	97.9	96.7	98.1	**98.4**
UPC-TALP	94.3	95.7	**96.7**	96.3	96.1	**96.7**
Freiburg-106	94.5	97.6	98.1	98.1	97.5	**98.6**
NAR	92.6	96.6	97.5	97.1	96.7	**97.8**

ITC-Irst, Freiburg-106, and NAR datasets, respectively. The improvements by the soft **BoP-1** systems are even better, reaching 0.6%, 1.8%, and 2.5%, respectively. There is only an exception on the UPC-TALP dataset on which the hard and soft **BoP-1** cause the accuracies to drop by 0.9% and 0.1% compared to the **BoW** baselines. It is also worth emphasizing that the dimensionality of the **BoP**s is far smaller than that of the best BoW competitors.

5.3.4.3 Performance of BoR systems

The performances of the classification systems using the BoR features are summarized in Table 5.8. The fusion system described in Section 5.2.3 is denoted as **BoR+**. As can be seen from the table, for the **BoR** systems, the χ^2 kernel is again found to be the most appropriate one, which yields the top accuracies on the UPC-TALP, Freiburg-106, and NAR datasets, respectively. Very good performances can also be seen with the linear kernel, which particularly produces the best result on the ITC-Irst dataset. Furthermore, as expected, by stacking responses of individual regressors in a bank to encode the shared features between different classes, the **BoR** systems significantly boost the classification performance to a higher level compared to the simple winner-take-all voting strategy in the **BoR-MV** system. Average absolute improvements of 2.4%, 2.4%, 3.6%, and 5.3% are obtainable for the ITC-Irst, UPC-TALP, Freiburg-106, and NAR datasets, respectively.

Compared to the systems based on the BoR features alone, the fusion system **BoR+** leads to improvements of 0.3%, 0.5%, and 0.3% over the **BoR** with χ^2 kernel. Nevertheless, no improvement is seen for the UPC-TALP dataset. While

the used fusion scheme is simple, other alternatives can also be sought, such as the multiple kernel learning framework [78].

5.3.4.4 Performance comparison

In order to justify the efficiency of the proposed classification systems, a comprehensive performance comparison on the experimental datasets is shown in Table 5.9. The table not only provides the comparison between the implemented systems' performances but also the results reported in previous works on the experimental datasets.

For the ITC-Irst dataset, three classification systems submitted to the CLEAR 2006 challenge [214] (i.e. *UPC-C*, *CMU-C1*, and *ITC-C1*) are used as competitors.

- *UPC-C* employed the set of low-level features similar to those in Section 3.3.3. Several statistics, including mean, standard deviation, entropy and autocorrelation coefficients, were further calculated to extract a global feature vector for each event instance. The classifier was trained using SVM with an RBF kernel.

- *CMU-C1* implemented a class-wise HMM recognizer for each target sound class. A set of 15 MFCCs was used as per-frame features. The recognizers were based on continuous density HMMs with customized topologies which were induced using the k-variable k-means algorithm [186]. Moreover, three complete sets of class-wise HMMs were learned and their scores were finally combined for the final recognition stage.

- *ITC-C1* was also based on class-specific recognizers using continuous density HMMs as in the *CMU-C1* system. Twelve MFCCs and the log-energy as well as their first and second derivatives were employed as per-frame features. Each event class was described by a three-state HMM with the left-to-right topology. All the HMMs used the output probability densities represented by means of 32 Gaussian components with diagonal covariance matrices. The training procedure was accomplished using the standard Baum-Welch algorithm [57, 235].

Most attempts on the UPC-TALP dataset employed class-specific continuous density HMMs. However, the dataset includes both multi-channel audio and video

Table 5.9: Classification performance comparison: (a) ITC-Irst, (b) UPC-TALP, (c) Freiburg-106, and (d) NAR.

(a)

System	Acc. (%)
BoW	96.4
PBoW-2	96.7
PBoW-3	95.9
PBoW-4	94.0
BoR-MV	95.5
hard **BoP-1**	**96.8**
hard **BoP-2**	**96.8**
hard **BoP-3**	96.2
soft **BoP-1**	**97.0**
soft **BoP-2**	**97.3**
soft **BoP-3**	**97.1**
BoR	**97.9**
BoR+	**98.4**
UPC-C [214]	95.6
CMU-C1 [214]	92.5
ITC-C1 [214]	87.7

(b)

System	Acc. (%)
BoW	96.3
PBoW-2	96.5
PBoW-3	96.6
PBoW-4	96.2
BoR-MV	94.3
hard **BoP-1**	95.4
hard **BoP-2**	95.8
hard **BoP-3**	96.0
soft **BoP-1**	96.2
soft **BoP-2**	**97.0**
soft **BoP-3**	**97.2**
BoR	**96.7**
BoR+	**96.7**
HMM+GMM [151]	87.6
AST [24]	90.7
AST+L+V [24]	92.9

(c)

System	F1-score (%)
BoW	95.9
PBoW-2	96.0
PBoW-3	95.8
PBoW-4	95.2
BoR-MV	92.4
hard **BoP-1**	97.6
hard **BoP-2**	97.5
hard **BoP-3**	97.8
soft **BoP-1**	98.2
soft **BoP-2**	**98.8**
soft **BoP-3**	**98.8**
BoR	97.6
BoR+	98.1
NEV [205]	92.0
DNN [97]	97.6
CNN [97]	98.3

(d)

System	Acc. (%)
BoW	94.5
PBoW-2	95.8
PBoW-3	96.4
PBoW-4	96.1
BoR-MV	92.6
hard **BoP-1**	95.3
hard **BoP-2**	94.7
hard **BoP-3**	93.3
soft **BoP-1**	**97.1**
soft **BoP-2**	**97.6**
soft **BoP-3**	**97.6**
BoR	**97.5**
BoR+	**97.8**
MFCC+TTFF+Interp [139]	97.0

recordings which have been explored using multi-modal approaches. The following systems of prior works are used for comparison in Table 5.9(b).

- *HMM* [151] was developed to jointly deal with event classification and localization. The spatial information outputted by localization was subsequently used to enhance the classification task. 16 *FFBCs* described in Section 3.3.3 with their first temporal derivatives were employed as per-frame features. The HMMs were left-to-right with three states for each event class. 32 Gaussian components with diagonal covariance matrices were used per state. Multi-channel audio data were also utilized. For multi-channel fusion, the likelihoods of the HMMs were fused with a maximum-a-posteriori (MAP) criterion [29].

- *AST+L+V* [24] is a multi-modal system in which audio, video, and spatial information (localization) modalities were combined. The same feature set as in the above *HMM* system was employed for the audio data. The spatial information was obtained using the SRP-PHAT localization method [53]. Several visual-based features were extracted for the video data, including person tracking features, motion history energy (MHE) features, object detection features, and door activities features [24]. The features were then concatenated and used as inputs for class-specific HMMs which are similar to those used in the above *HMM* system. Although results with different combinations of the three modalities were reported in [24], only the one with audio data (i.e. *AST*) and the one with all modalities combined (i.e. *AST+L+V*) are included in Table 5.9(b) for clarity.

The results on the Freiburg-106 dataset have been reported in the following works:

- *NEV* [205] presented a non-Markovian ensemble voting approach for classification. The idea is to use the local feature vectors to vote for the center of a target event. The events were first decomposed into multiple frames each of which is represented by twelve MFCCs. During training, a frame-wise random-forest classifier [18] was trained and a codebook was learned for each event class using k-means clustering. During testing, a frame-wise local feature vector is firstly classified by the classifier. It is then matched to a cluster of the corresponding class-specific codebook. The members of the

matched cluster are finally used to vote for the target event center. The votes made by all local feature vectors of the target event were accumulated and the classification label was decided by the maximum score.

- *DNN* and *CNN* were recently proposed in the work of Hertel et al. [97] which compared the performance of deep networks on the audio event classification task using waveform and time-frequency inputs. Only the networks trained on time-frequency inputs are considered for comparison here since they are superior to the counterparts trained on the raw waveform [97]. Both the DNN and CNN were trained and evaluated on 150-ms-long segments, followed by probability voting for a global classification label [178]. The DNN comprises more than 1.5 million trainable weights with six fully-connected layers coupled with dropout layers. The CNN has four convolutional-pooling layers and three fully-connected ones coupled with dropout layers, consisting of nearly two million trainable parameters.

Turning to the NAR dataset, in the seminal work [139], the authors benchmarked different low-level features (e.g. MFCC, time and time-frequency features (TTFF), wavelets, and stabilized auditory images (SAI)) combined with various post-processing techniques to integrate successive frame-wise feature vectors, including temporal concatenation, average pooling, bag-of-words modeling, and interpolation. The classification was accomplished with various back-end classifiers, including kNN, Quantized Nearest Neighbor (QNN), GMM, HMM, and SVM. The system *MFCC+TTFF+Interp* which uses the combination of MFCC and TTFF features, followed by the interpolation post-processing and SVM classification yielded the best accuracy at 97.0%. Only this system is included for comparison in Table 5.9(d).

In Table 5.9, to make a proper comparison, the results on the Freiburg-106 dataset are reported on F1-score as in the previous works in [97, 205]. As can be seen from the table, most of the time the proposed systems are superior to all the competitors, i.e. the developed baselines and the previous reported results. On the ITC-Irst, UPC-TALP, Freiburg-106, and NAR datasets, the best systems (i.e. **BoR+**, soft **BoP-3**, soft **BoP-2**, and **BoR+**) outperform the best baselines (i.e. **PBoW-2**, **PBoW-3**, **PBoW-2**, and **PBoW-3**) by 1.7%, 0.8%, 2.8%, 1.4% absolute and the best previously reported results (i.e. *UPC-C*, *AST+L+V*, *CNN*, and *MFCC+TTFF+Interp*) by 2.8%, 4.3%, 0.5%, 0.8% absolute, respectively. Noticeably, for the Freiburg-106 dataset, the **BoR+**'s and the soft **BoP-1**'s

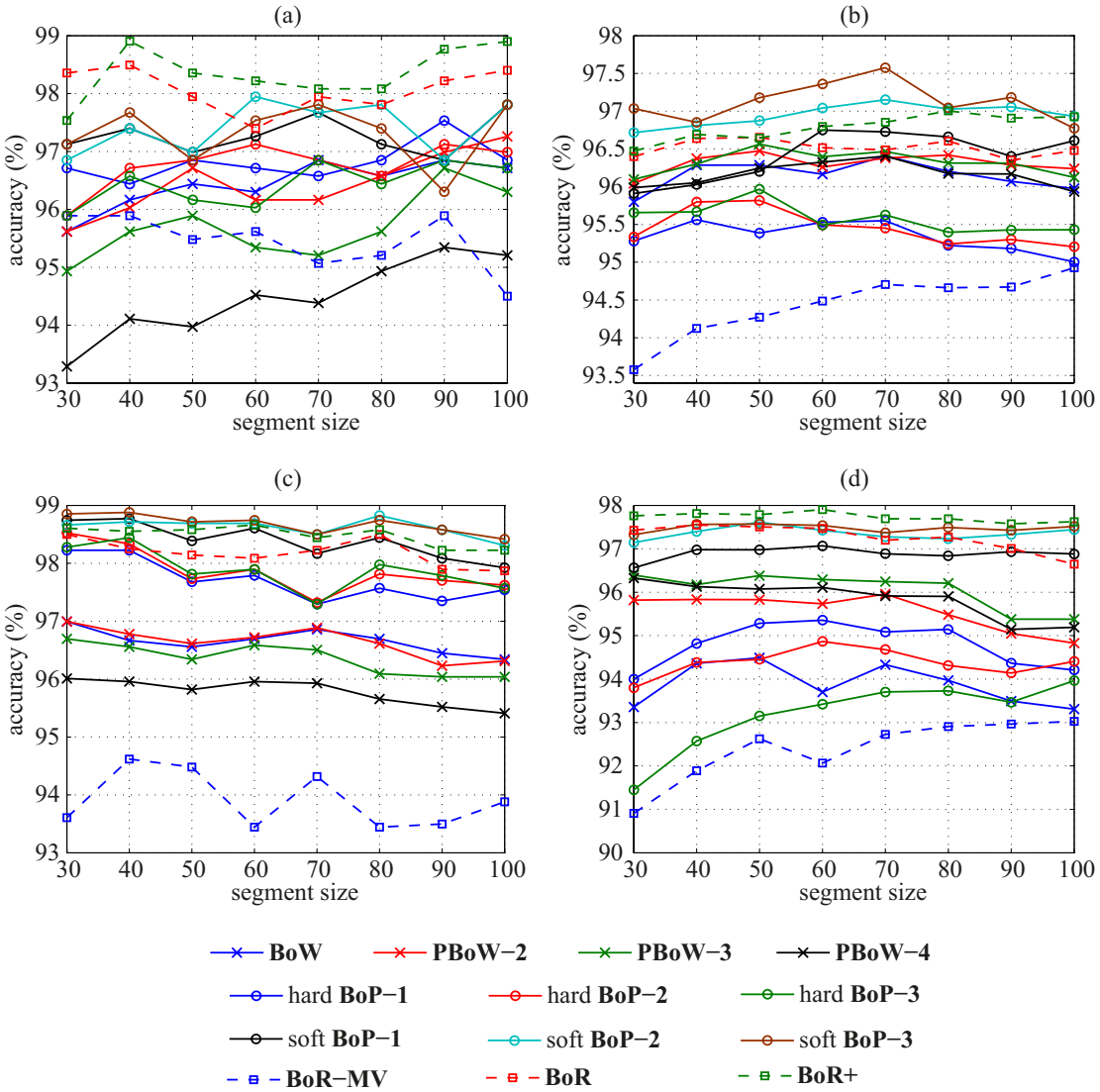

Figure 5.6: The accuracies of different classification systems varying as functions of the segment size. (a) ITC-Irst, (b) UPC-TALP, (c) Freiburg-106, and (d) NAR.

performances are on par to that of the deep *CNN* [97] with millions of parameters despite their extremely low dimensionality. The soft **BoP-2** and **BoP-3** are even better, outperforming the *CNN* by 0.5% absolute.

5.3.5 Effects of the Audio Segment Size

The influence of the audio segment size is investigated in this section. This analysis is accomplished by varying the segment size in the range $\{30, 40, \ldots, 100$ ms$\}$. For

each size, the experiments in the previous section were repeated with other settings unchanged. Figure 5.6 shows variations of the classification accuracies of different implemented systems as functions of the segment size. For all the experimental datasets, the performance curves of the soft BoP-based and BoR-based systems remain above those of the baselines, indicating that their superior performances are invariant to the different studied segment sizes. Furthermore, their performance curves fluctuate much less than those of the baselines, except for the ITC-Irst dataset, which can be explained by its relatively small test set. Taking the **BoR+** for instance, the standard deviations of its accuracies on the ITC-Irst, UPC-TALP, Freiburg-106, and NAR datasets are only 0.5%, 0.2%, 0.2%, and 0.1%, respectively.

6 Speech-Based Generic Representations of Audio Event Classification

In the domain of computational analysis of nonspeech audio signals, signal representation remains a fundamental problem for many successive tasks such as classification and detection. Thus far, many works have focused on the efficiency of signal representations in terms of maximizing performance of the classification or detection task at hand [39, 41, 49, 97, 125, 162, 178, 180, 236]. The learned representations presented in Chapter 5 also concentrated on this perspective. Although considerable progress has been made in different benchmark datasets, more often than not, these representations are data-specific. The community still lacks a generic representation that is learned to characterize audio events from different data sources in a homogeneous manner. Such a generic representation can provide a unified framework to cope with audio events. In this chapter, a generic representation, which exploits speech patterns as basic elements to characterize nonspeech audio events, will be presented.

6.1 Overview

Inspired by the fact that the human auditory system is very well matched to both human speech and environmental sounds, the question arises whether human speech material may provide useful information for training systems for analyzing nonspeech audio signals, for example, in a classification task. In order to answer this question, in the proposed generic representation, different speech patterns are considered as basic acoustic concepts which embed and represent target nonspeech audio event instances by measuring their similarity to the speech patterns. It should be emphasized that the speech patterns are obtained from an external independent source which is totally unrelated to the target audio events. By collecting a sufficiently large set of speech patterns, it is expected to cover a wide range of acoustic concepts of the world whose combinations can represent

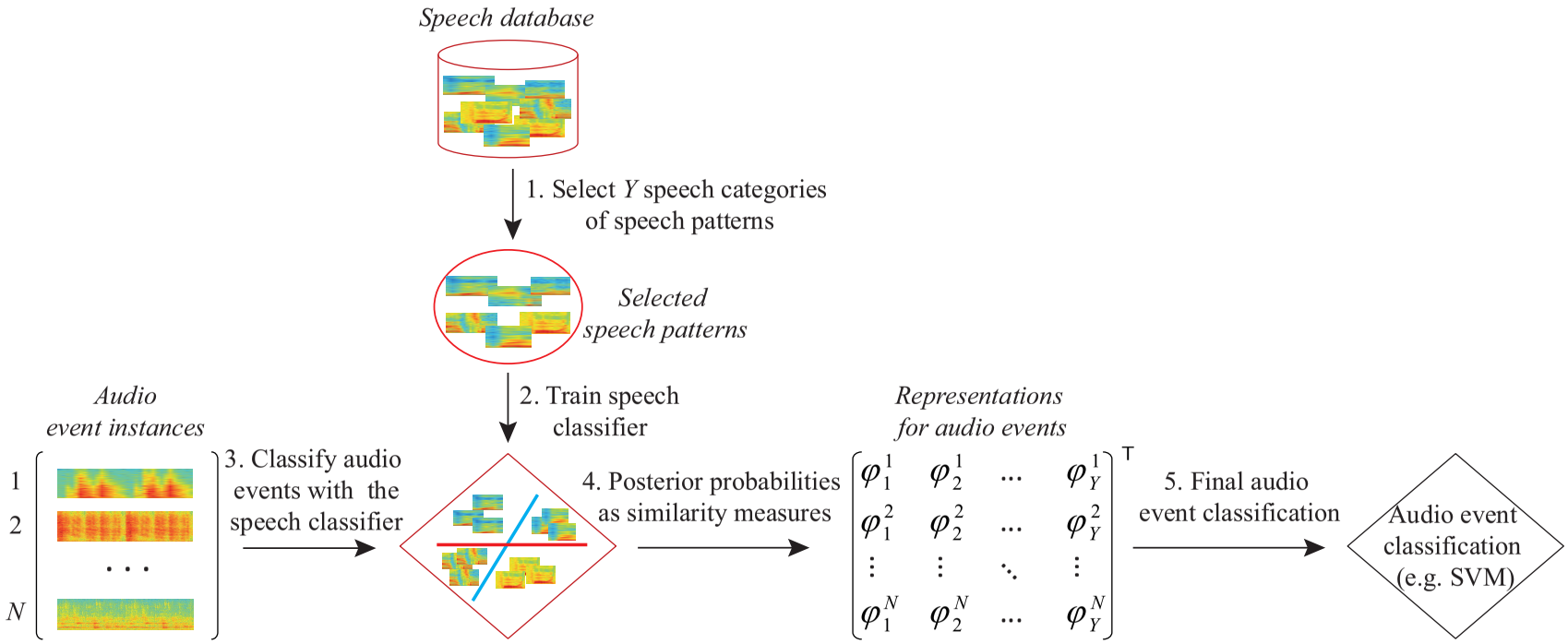

Figure 6.1: The overall idea. It can be summarized in five steps. 1) Given a speech database, Y speech categories are extracted and used as speech patterns. 2) A speech classifier is learned to classify the selected speech patterns. 3) N target nonspeech audio event instances are then classified using the learned speech classifier. 4) The i-th target event, where $i \in \{1, 2, \ldots, N\}$, is represented by the i-th vector $\begin{pmatrix} \varphi_1^i & \varphi_2^i & \ldots & \varphi_Y^i \end{pmatrix}^\mathsf{T} \in [0,1]^Y$ of the matrix whose entries are the resulted classification posterior probabilities. The entry φ_j^i of this vector indicates the similarity between the i-th target event instance and the speech pattern $j \in \{1, 2, \ldots, Y\}$. 5) The final audio event classification can be accomplished, e.g. by an SVM classifier, using the speech-based features.

different kinds of sounds. As a result, embedding the target audio events into the space spanned by the similarities to these concepts is expected to produce a good representation.

To accomplish this, given a set of labeled speech samples (e.g. speech words or phones) of different categories, these speech categories are employed as speech patterns. As the two terms "speech category" and "speech pattern" refer to the same thing, they will be used interchangeably. A multi-class speech classifier is then trained to separate the speech patterns. Given a target audio event instance, it is presented to the speech classifier to obtain a vector of classification posterior probabilities with which the target event is classified into different speech patterns. These posterior probabilities can be interpreted as the *similarities* of the target event and the speech patterns. The vector of posterior probabilities is then used as a descriptor for the target event and the speech classifier, therefore, plays the role of a feature extractor. The idea is illustrated in Figure 6.1. The speech-based descriptors extracted like this are generic in the sense that once the feature extractor, i.e. the speech classifier, has been trained, it can be used to extract features for any input audio event without re-training. This is opposed to other common feature learning methods, such as those in [105, 125, 162, 180] as well as those described in Chapter 5, which are usually optimized for a specific target dataset.

An improved method will be further presented to automatically organize the speech patterns hierarchically with a label tree. The label tree is learned to recursively group similar speech patterns into clusters along the tree so that the speech meta-classes (i.e. the speech clusters) can be easily separated from one another. The intention of doing this is to gain the distinctiveness between the speech meta-classes which is then expected to improve the representation capability (i.e. the audio event classification accuracy) of the original set of speech patterns. Afterwards, multiple binary meta-class classifiers are trained and associated at the split nodes of the tree. A label tree embedding is finally derived using these classifiers. Via the embedding, the target audio event is mapped to and represented by the likelihoods with which it is classified into different speech meta-classes by the binary classifiers.

In addition, a selection algorithm is also introduced to extract a *sufficient subset* of speech patterns from the original entire set for representation learning purpose. This subset can closely approximate the representation capability of the entire set,

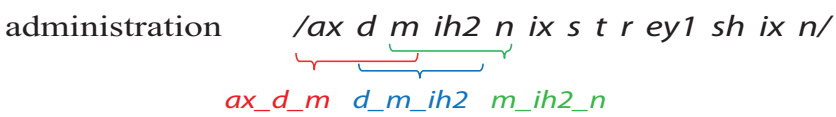

Figure 6.2: An example of phone triplets. The word "administration" is decomposed into its constituent monophones. Phone triplets, such as `ax_d_m`, `d_m_ih2`, and `m_ih2_n`, are combinations of three consecutive phones.

yet its size is significantly smaller. As a result, it is computationally more efficient for both the representation learning and final event classification stages than the original set.

6.1.1 Speech Patterns

There exist different speech levels (e.g. phone, word) that may be considered for speech patterns. Whereas the number of single phones is limited, combining them would create more diverging speech patterns, and hence enriches the representation. *Phone triplet*, a combination of three successive speech phones, are proposed for this purpose. Some examples of phone triplets are demonstrated in Figure 6.2. Note that phone triplets are different from triphones that have been commonly adopted in the speech recognition task [68, 127, 149, 238]. A triphone is a single phone that takes into account the previous and successive phones as the context. In contrast, a triplet is a combination of three consecutive phones as a whole. Furthermore, the temporal order of the constituent phones in a triplet is considered not important. For example, all combinations of three single phones {`ax`, `d`, `m`}, such as `d_ax_m`, `ax_d_m`, and `m_ax_d`, etc. are defined to belong to the same category. It will be studied in Section 6.4.3.4 how retaining the order of the constituent phones will affect the final audio event classification performance.

There are other reasons why using phone triplets here would be more appropriate than the short phone units. Nonspeech audio events are usually long signals (in the order of hundreds of milliseconds up to several seconds), which are much longer than phone units. Therefore, phone triplets, which are longer speech segments than the phone units (i.e. a phone triplet is about three times longer than a triphone), are more appropriate for long nonspeech audio event signals than the single phones alone. Higher orders of phone combination would also be appropriate, such as speech words, but they require more data to ensure a good coverage, which are not always available. A comparative study will be conducted in Section 6.4.3.4 to

demonstrate that phone triplets result in better descriptors for audio events than those obtained with speech words extracted from the same speech database.

6.1.2 Employed Low-Level Features

In order to measure the similarities between a target nonspeech audio event and speech patterns, the speech and nonspeech signals are necessary to be represented by common low-level features. The signals were firstly downsampled to 16 kHz. Each of them is then decomposed into multiple segments. The set of 53 low-level features described in Section 3.3.3 was utilized to characterize each segment. In turn, a whole signal, either a phone triplet or an audio event, is represented by a 106-dimensional feature vector computed by the mean and standard deviation over its per-segment feature vectors.

A segment length of 50 ms with a step size of 10 ms was used for audio event decomposition whereas a segment length of 25 ms was used for phone triplets. For speech, a segment length of 20-30 ms is common because the signal in a segment is more or less stationary, and the shortest phones (e.g. some plosives) have a duration of around 20 ms. In contrast, nonspeech audio events exhibit a wider range of characteristics [48, 73]. Moreover, for the audio event classification task it is important to recognize an audio event as a whole and not every single 20 ms fragment of it. Therefore, longer segments appear to be more reasonable for audio events than conventional short ones.

6.2 Generic Speech-Based Descriptors for Nonspeech Audio Events

In the section, two types of generic speech-based descriptors for nonspeech audio events will be presented: the *flat* descriptor in Section 6.2.1 and the *tree-induced* descriptor in Section 6.2.2, that are built on top of the low-level features. The former is derived using a single multi-class speech pattern classifier. The latter is extracted using a label tree embedding whose ingredients are binary speech meta-class classifiers hierarchically associated with the label tree.

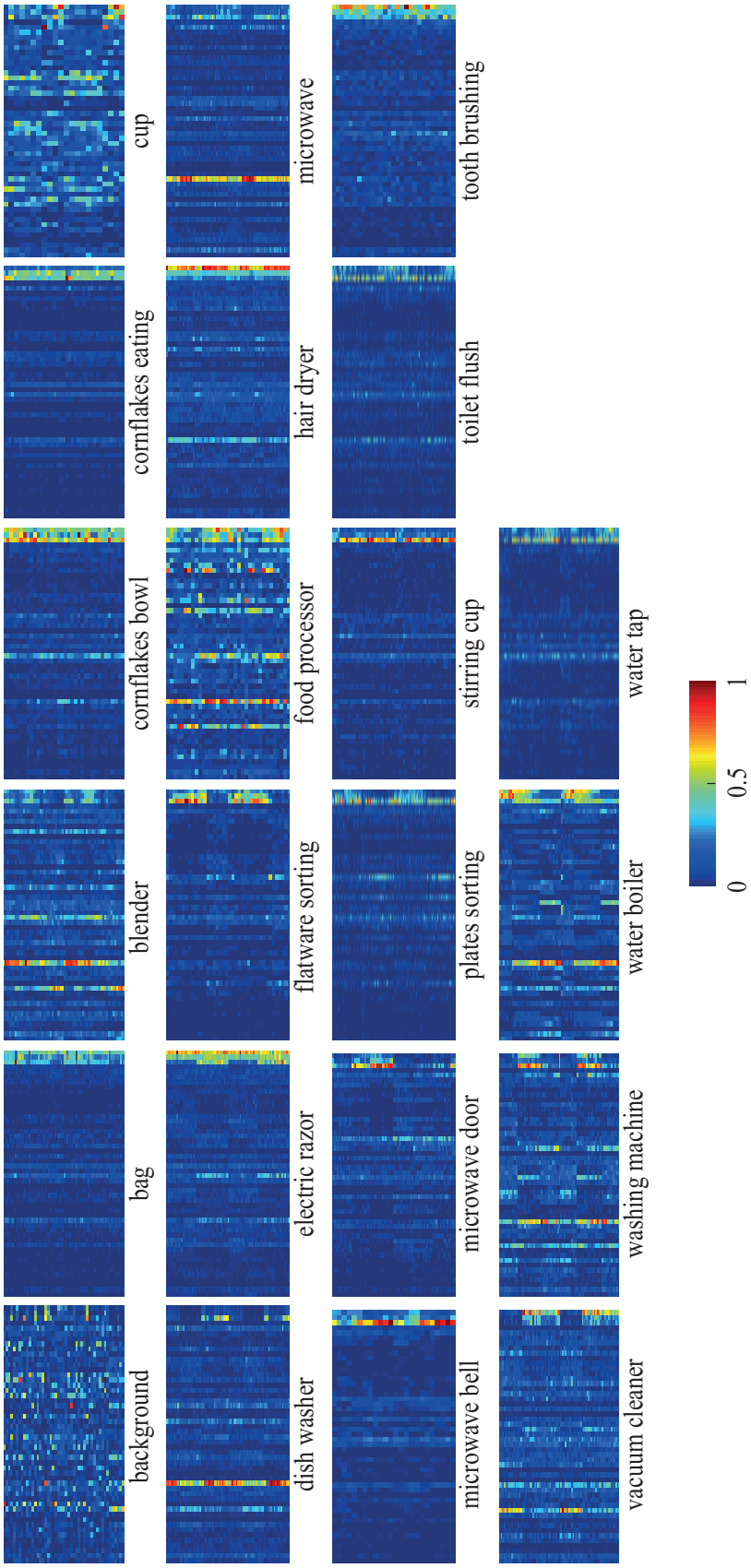

Figure 6.3: Similarities between audio events of different categories of the Freiburg-106 dataset and 50 phone triplet categories of the TIMIT dataset. The phone triplets are randomly selected. On an image corresponding to an event category, the event instances are indexed in the rows while the phone triplet categories are indexed in the columns. Each row of the image, therefore, contains the similarities of the respective event instance to the 50 phone triplet categories.

6.2.1 Flat Descriptor via Speech Pattern Similarities

Let $\mathcal{Z} = \{(\mathbf{z}_i, y_i); i \in \{1, 2, \ldots, N_{\mathcal{Z}}\}\}$ denote a phone triplet set that will be used for learning audio event representations. The set consists of $N_{\mathcal{Z}}$ phone triplets of Y categories. The variable $\mathbf{z}_i \in \mathbb{R}^D$ and $y_i \in \{1, 2, \ldots, Y\}$ denotes the low-level feature vector and the class label of the i-th sample, respectively. The feature vector $\mathbf{z}_i$ contains the $D = 106$ low-level features as described in Section 6.1.2.

Given a target audio event characterized by the low-level feature vector $\mathbf{x} \in \mathbb{R}^D$, the goal is then to represent it in terms of its similarities to Y phone triplet categories in the set $\mathcal{Z}$. This is accomplished in two steps. Firstly, a speech classifier $\mathcal{M}_{\mathcal{Z}}$ is trained to separate Y phone triplet categories in $\mathcal{Z}$. Secondly, the target event is presented to the classifier to obtain the classification posterior probability vector:

$$\boldsymbol{\varphi}(\mathbf{x}) \;=\; \Big(\; \varphi_1(\mathbf{x}) \quad \varphi_2(\mathbf{x}) \quad \ldots \quad \varphi_Y(\mathbf{x}) \;\Big)^{\mathsf{T}} \in [0,1]^Y, \tag{6.1}$$

where

$$\varphi_i(\mathbf{x}) = P(i \,|\, \mathbf{x}) \text{ for } i \in \{1, 2, \ldots, Y\}. \tag{6.2}$$

Each entry $\varphi_i(\mathbf{x})$ quantifies how likely the target event is classified as the phone triplet category $i \in \{1, 2, \ldots, Y\}$. Thus, it can be interpreted as a similarity measure. The vector $\boldsymbol{\varphi}(\mathbf{x})$ is finally used to represent the target audio event.

The multi-class phone triplet classifier $\mathcal{M}_{\mathcal{Z}}$ is trained using random-forest classification [18]. Other classification algorithms can also be suitable for this purpose, e.g. DNNs. Traditionally, the obtained posterior probabilities are used for decision making, such as in a recognition task. Here, they are used as features to represent the target audio event. The classifier $\mathcal{M}_{\mathcal{Z}}$, therefore, plays the role of a feature extractor. As a result, the target event is embedded in the space spanned by its similarities to the phone triplet categories in the set $\mathcal{Z}$.

For illustration, Figure 6.3 shows the similarities between audio event instances of different categories of the Freiburg-106 dataset [205] and 50 phone triplet categories of the TIMIT dataset [64]. The phone triplet categories are randomly selected among 2,256 available categories. In this example, the random-forest classifier $\mathcal{M}_{\mathcal{Z}}$ consists of 200 trees. Further details will be described in the experiments in Section 6.4. As can be seen from the figure, different event categories exhibit

distinguished similarity patterns, except for the "background" class. This class shows random responses since it covers different kinds of sounds.

6.2.2 Tree-Induced Descriptor via a Speech Label Tree Embedding

It is intuitive that in order to learn a good descriptor for a target audio event, the speech patterns in the set $\mathcal{Z}$ should be as distinctive as possible from one another. Armed with expertise, one can carefully select such speech categories manually from a speech database. Here, given a pre-determined set of speech patterns, the goal is to explore the structure of the speech labels to form distinctive speech meta-classes automatically. The idea is to recursively cluster the speech categories into meta-classes in such a way that they are easy to be distinguished from one another. Towards this end, a label tree is learned for the speech categories similar to that in [14]. When the label tree is constructed, different binary meta-class classifiers can be trained and associated with the split nodes of the tree. As a result, the multi-class speech classification problem in Section 6.2.1 is reduced to multiple binary classification problems. The ensemble of the binary classifiers is then used to derive the label tree embedding which transforms the target audio event into the meta-class likelihoods for representation. Furthermore, due to the way the meta-classes are formed, the average classification accuracy obtained by the binary classifiers is much better than that of the flat multi-class classifier. This will be shown in Section 6.4.3.5 to be an important factor to achieve good representations for nonspeech audio events.

6.2.2.1 Speech label tree construction

Let $\mathcal{L} = \{1, \ldots, Y\}$ denote the label set of the phone triplet set $\mathcal{Z}$. The label tree is constructed in a recursive manner so that each of its nodes is associated with a subset of the entire set $\mathcal{L}$. The learning algorithm starts with the root node which is linked to $\mathcal{L}$. Without loss of generality, let us consider a current split node with a label subset $\ell \subset \mathcal{L}$. The aim is to split ℓ into two smaller subsets ℓ^{L} and ℓ^{R} that fulfill the following conditions: $\ell^{\mathrm{L}} \neq \emptyset$, $\ell^{\mathrm{R}} \neq \emptyset$, $\ell^{\mathrm{L}} \cup \ell^{\mathrm{R}} = \ell$, and $\ell^{\mathrm{L}} \cap \ell^{\mathrm{R}} = \emptyset$.

Among $2^{|\ell|-1} - 1$ such possible partitions $\{\ell^L, \ell^R\}$, the optimal one is adopted such that ℓ^L and ℓ^R can be separated with as few errors as possible using a binary classifier. An exhaustive search for such an optimal partition would be prohibitively

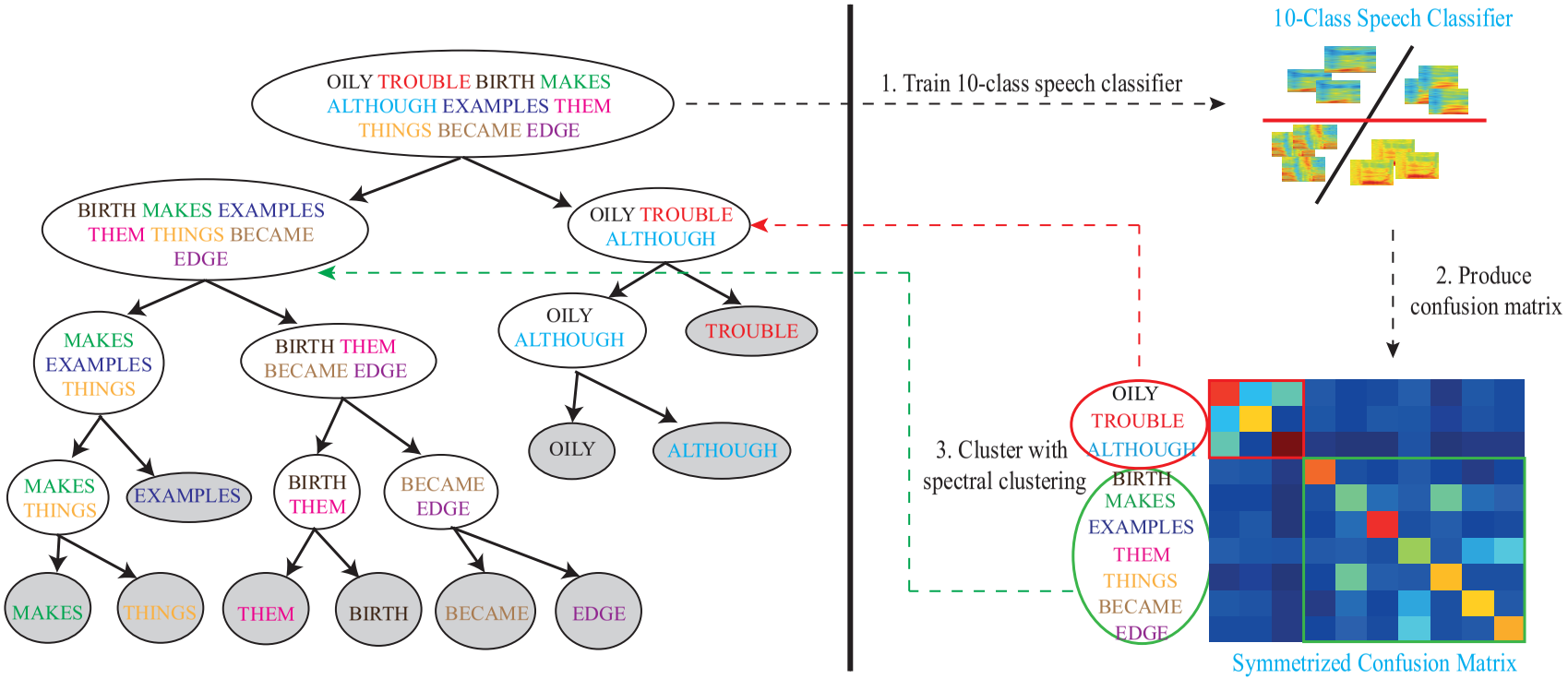

Figure 6.4: Illustration of label tree construction. The left side shows the constructed label tree for the ten selected TIMIT word categories using the algorithm described in Section 6.2.2.1. The white and shaded nodes represent the split and leaf nodes, respectively. On the right, the splitting process at a split node (the root) is illustrated: (1) the multi-class speech classifier is learned, (2) the confusion matrix is then obtained, and (3) two clusters of speech labels are formed using spectral clustering and assigned to the child nodes. The words in the same cluster usually contains similar phones, for example the two closely related phones `l` (in "OILY" and "ALTHOUGH") and `el` (in "TROUBLE").

expensive especially when $|\mathcal{L}|$ is large. Alternatively, a multi-class classification confusion matrix can be leveraged for this purpose. This matrix indicates how well a class is separated from the others. As a result, those classes that tend to be confused with each other should be grouped into the same cluster.

Let $\mathcal{Z}^\ell \subset \mathcal{Z}$ denote the subset of phone triplets corresponding to the label set ℓ, i.e. $\mathcal{Z}^\ell = \{(\mathbf{z}, y) \in \mathcal{Z} \,|\, y \in \ell\}$. Furthermore, suppose that $\mathcal{Z}^\ell$ has been divided into two equal halves: $\mathcal{Z}^\ell_{\text{train}}$ for training a classifier and $\mathcal{Z}^\ell_{\text{eval}}$ for evaluation. A multi-class classifier $\mathcal{M}^\ell_{\mathcal{Z}}$ is then trained using the phone triplets of the set $\mathcal{Z}^\ell_{\text{train}}$. Again, random-forest classification [18] is employed for training purpose. Subsequently, the classifier $\mathcal{M}^\ell_{\mathcal{Z}}$ is exercised on the phone triplets of the evaluation set $\mathcal{Z}^\ell_{\text{eval}}$ to obtain the confusion matrix $\mathbf{A} \in \mathbb{R}^{|\ell| \times |\ell|}$. Each element $\mathbf{A}_{ij}$ of the matrix $\mathbf{A}$ is computed by

$$\mathbf{A}_{ij} = \frac{1}{|\mathcal{Z}^\ell_{\text{eval}}(i)|} \sum_{(\mathbf{z},y) \in \mathcal{Z}^\ell_{\text{eval}}(i)} P(j \,|\, \mathbf{z}). \tag{6.3}$$

Here, $\mathcal{Z}^\ell_{\text{eval}}(i) \subset \mathcal{Z}^\ell_{\text{eval}}$ is the subset of phone triplets with respect to the label i, i.e. $\mathcal{Z}^\ell_{\text{eval}}(i) = \{(\mathbf{z}, y) \in \mathcal{Z}^\ell_{\text{eval}} \,|\, y = i \in \ell\}$, and $P(j \,|\, \mathbf{z})$ denotes the posterior probability with which the classifier $\mathcal{M}^\ell_{\mathcal{Z}}$ predicts the phone triplet represented by $\mathbf{z}$ as the class $j \in \ell$. The element $\mathbf{A}_{ij}$ implies how likely a phone triplet of the class i is predicted to belong to the class j by the classifier. Since $\mathbf{A}$ is not symmetric, it can be symmetrized as

$$\bar{\mathbf{A}} = (\mathbf{A} + \mathbf{A}^\mathsf{T})/2. \tag{6.4}$$

Eventually, the optimal partitioning $\{\ell^\text{L}, \ell^\text{R}\}$ is selected to maximize:

$$E(\ell) = \sum_{i,j \in \ell^\text{L}} \bar{\mathbf{A}}_{ij} + \sum_{k,l \in \ell^\text{R}} \bar{\mathbf{A}}_{kl}. \tag{6.5}$$

The intention of doing this is to group the ambiguous phone triplet categories into the same cluster and, as a result, produce two meta-classes $\{\ell^\text{L}, \ell^\text{R}\}$ that are expected to be easily separated from each other. Since it is hard to solve the optimization problem in Eq. (6.5) directly, spectral clustering [155] is applied on the matrix $\bar{\mathbf{A}}$ to solve a relaxed version of it. The label subsets ℓ^L and ℓ^R are then directed to the left and right child nodes of the current split node, respectively.

The splitting process is recursively repeated to grow the whole label tree. It is terminated when a leaf node with a single class label is reached.

Figure 6.4 portrays a simplified example in which a label tree is constructed for ten randomly selected speech categories of the TIMIT dataset. For the sake of clarity, in this demonstration, speech word categories are employed for speech patterns instead of phone triplets.

6.2.2.2 Label tree embedding for tree-induced descriptor

After being constructed, the label tree comprises $(Y-1)$ split nodes in total. Furthermore, the original label set $\mathcal{L}$ has been divided into $(Y-1)\times 2$ subsets which are considered as meta-classes. Two of the meta-classes are associated with the left and right child nodes of a split node in the tree. The objective is then to transform the target audio event represented by $\mathbf{x}\in\mathbb{R}^D$ into the vector whose entries are the meta-class likelihoods for representation.

Suppose that the split nodes are indexed as $\{1,2,\ldots,Y-1\}$. Formally, the explicit label tree embedding $\Psi : \mathbb{R}^D \to [0,1]^{(Y-1)\times 2}$ is derived to map the target event to the feature vector

$$\Psi(\mathbf{x}) = \begin{pmatrix} \psi_1^{\mathrm{L}}(\mathbf{x}) & \psi_1^{\mathrm{R}}(\mathbf{x}) & \psi_2^{\mathrm{L}}(\mathbf{x}) & \psi_2^{\mathrm{R}}(\mathbf{x}) & \ldots & \psi_{Y-1}^{\mathrm{L}}(\mathbf{x}) & \psi_{Y-1}^{\mathrm{R}}(\mathbf{x}) \end{pmatrix}^{\mathsf{T}}. \quad (6.6)$$

The entries $\psi_i^{\mathrm{L}}(\mathbf{x})$ and $\psi_i^{\mathrm{R}}(\mathbf{x})$ denote the likelihoods with which the target event $\mathbf{x}$ belongs to two meta-classes on the left and right child nodes of the split node $i\in\{1,2,\ldots,Y-1\}$, respectively. To compute the likelihoods, at the split node i associated with the label subset $\ell_i\subset\mathcal{L}$ and the optimal partition $\{\ell_i^{\mathrm{L}},\ell_i^{\mathrm{R}}\}$, a binary classifier $\mathcal{M}_{\mathcal{Z}}^{\ell_i}$ is trained similar to the label-tree construction stage. The only exception is that the whole phone triplet subset $\mathcal{Z}^{\ell_i}\in\mathcal{Z}$ corresponding to the label subset ℓ_i is used as the training data here. For training purpose, the phone triplets with their labels in ℓ_i^{L} and ℓ_i^{R} are considered as negative and positive examples, respectively. The likelihoods $\psi_i^{\mathrm{L}}(\mathbf{x})$ and $\psi_i^{\mathrm{R}}(\mathbf{x})$ then read

$$\psi_i^{\mathrm{L}}(\mathbf{x}) = P(\text{negative}\,|\,\mathbf{x}), \quad (6.7)$$

$$\psi_i^{\mathrm{R}}(\mathbf{x}) = P(\text{positive}\,|\,\mathbf{x}). \quad (6.8)$$

Here, $P(\text{negative}\,|\,\mathbf{x})$ and $P(\text{positive}\,|\,\mathbf{x})$ are the classification posterior probabilities outputted by the classifier $\mathcal{M}_{\mathcal{Z}}^{\ell_i}$ when evaluating the target event $\mathbf{x}$, thanks to the probability support of the random-forest classification [18].

6.2.3 Discussion on Speech Patterns Classifiers

In the field of ASR, it is well known that the temporal dynamics are useful for speech modeling [187]. Although it appears plausible to employ the conventional speech models for speech classification in Sections 6.2.1 and 6.2.2, e.g. frame-based acoustic modeling followed by temporal sequencing by HMMs [187], there are various reasons for not doing so. Firstly, human speech is temporally well-structured, meaning that it is easy to decompose it into constituent phones. As a result, HMMs that explicitly model temporal dependencies are able to capture the development of the speech signals very efficiently. In contrast, the characteristics of nonspeech audio events significantly differ from those of speech. That is, no sub-word dictionary exists for the audio events, and compared to speech, they expose a much wider variety in frequency content, duration, and profile. As a result, an HMM that assumes a first-order Makov process is likely to be inefficient to capture the temporal information of the audio events. In practice, ASR-like systems have been shown to be inferior to classifiers trained on global features extracted from the whole signals for the classification task [75, 214, 215]. These arguments lead to the choice of the simple random-forest classifiers trained on global features of the speech signals during the presentation learning process.

6.3 Selection Algorithm for a Sufficient Subset of Speech Patterns

Intuitively, among a large number of speech patterns, some are more representative for a specific category of target audio events, and therefore, more contributive to the representations of these events than others. If the significantly contributive speech patterns can be somehow selected, they should be sufficient to represent the target audio events. In contrast, the others, which have negligible influence to the representations, can be discarded. This selection, if possible, will gain different benefits. Firstly, the computational overhead of the representation learning process can be reduced. Second, the dimensionality of the obtained speech-based descriptors can be diminished, and as a result, brings down the computational cost for training and evaluating the final event classifiers.

Moreover, a good representation, in general, should bring the event instances belonging to the same class close together in the feature space while keeping them

far away from the samples of other classes. Assume that we are able to measure the similarity between a speech pattern and an event category. In this sense, a speech pattern resulting in a flat similarity distribution over different target event categories would not enhance the representation capability in telling apart audio events of different categories. Thus, such a speech pattern should not be included. In contrast, a speech pattern that has a skewed similarity distribution, peaking on a certain event category, would gain the representation's discrimination power to separate audio events of this category from others. A simple method is presented in this section to select a small subset of discriminative speech patterns that is expected to be sufficient to represent audio events of a target dataset at hand.

Let the set of audio events $\mathcal{E} = \{(\mathbf{x}_i, c_i); i \in \{1, 2, \ldots, N_\mathcal{E}\}\}$ denote the target event dataset of $N_\mathcal{E}$ event instances and C categories. The variables $\mathbf{x}_i \in \mathbb{R}^D$ and $c_i \in \{1, 2, \ldots, C\}$ denote the low-level feature vector and the class label of the i-th sample, respectively. The feature vector $\mathbf{x}_i$ includes the low-level features as described in Section 6.1.2.

The selection method requires a way to quantify the similarity between a speech category $i \in \{1, 2, \ldots, Y\}$, i.e. a speech pattern, and a target event category $j \in \{1, 2, \ldots, C\}$. It can be accomplished in a similar, but reversed manner to the approach used to measure similarities between an audio event instance and different speech patterns in Section 6.2.1. Firstly, a multi-class event classifier $\mathcal{M}_\mathcal{E}$ is trained using the target event instances in $\mathcal{E}$ as the training data. Again, the classifier is trained with random-forest classification [18]. Secondly, being presented with a phone triplet represented by the feature vector $\mathbf{z} \in \mathbb{R}^D$, the similarity between the input phone triplet and the event category j can be obtained as $P(j \,|\, \mathbf{z})$. The term $P(j \,|\, \mathbf{z})$ is the classification posterior probability with which the input phone triplet is classified as the event category j by the classifier $\mathcal{M}_\mathcal{E}$. Eventually, the similarity $\nu(i, j)$ between the phone triplet category i and the event category j is computed as

$$\nu(i, j) = \frac{1}{|\mathcal{Z}(i)|} \sum_{(\mathbf{z}, y) \in \mathcal{Z}(i)} P(j \,|\, \mathbf{z}). \tag{6.9}$$

Here, $\mathcal{Z}(i) \subset \mathcal{Z}$ denotes the phone triplet subset with respect to the speech category i, i.e. $\mathcal{Z}(i) = \{(\mathbf{z}, y) \in \mathcal{Z} \,|\, y = i\}$.

After ranking the phone triplet categories based on their similarities to a certain event category, a sufficient subset of speech patterns for this event category can

be easily obtained. The subset is initialized with the speech pattern that has the highest similarity. The next T speech patterns with the most similarities then can be repeatedly included into the subset. These steps are performed simultaneously for all C target event categories. At each step, the flat or tree-induced descriptors are learned as in Sections 6.2.1 or 6.2.2, respectively, for the event instance using the current subset. The representation capability of the current subset can be evaluated via the cross-validation accuracy of the event classification. The selection procedure is continued as long as a better or equal cross-validation accuracy is achieved. It should be noted that nothing prevents a single speech pattern to be selected by several audio event categories. This is expected due to sharing features between them.

6.4 Experiments

This section will elaborate the experiments conducted on different audio event datasets to justify the efficiency of the generic speech-based descriptors for the event classification task. Different factors that influence the speech-based descriptors are also investigated.

6.4.1 Experimental Datasets

The speech TIMIT dataset [64] was employed as the independent speech database from which the phone triplet categories were extracted and used as speech patterns in the experiments. This database contains about five hours of speech with 6,300 utterances in total. Overall, 630 speakers from eight major dialect divisions of the United States spoke ten sentences. The phonetic set consisting of 61 phones was reduced to 39 base phones following the standard procedure [127]. With 39 base phones, there exists a vast number of possible triplet categories. However, in order to learn reliable phone triplet classifiers for representation learning as described in Sections 6.2.1 and 6.2.2, only those with at least ten samples were exploited. Consequently, 2,256 of such phone triplet categories were retained. Lastly, for each phone triplet category, at most 100 randomly selected samples were kept. The rational is to make an even distribution of training data among speech categories and, hence, a balanced classification problem.

Finally, the event instances of the four datasets ITC-Irst [214], UPC-TALP [24], Freiburg-106 [205], and NAR [139] used in the previous chapter were considered as target audio events.

6.4.2 Final Audio Event Classification

After obtaining the speech-based descriptors (i.e. the flat and tree-induced descriptors) for the target audio events, the final event classification systems were trained using one-vs-one SVMs. Four kernels were considered, including linear, RBF, χ^2, and hist. kernels. The hyperparameters of the SVMs were tuned via 10-fold cross-validation. The grid search for the hyperparameters was conducted as in Section 5.3.2.

The random-forest classifiers used for phone triplet classification described in Sections 6.2 and 6.3 were trained with the algorithm in [18] with 200 trees each. It will be discussed in Section 6.4.3.5 how varying the number of trees will affect the learned descriptors.

6.4.3 Experimental Results

The experimental results will be described in detail in this section. The classification performances obtained with the flat and tree-induced descriptors with different number of speech patterns are firstly elaborated. The representation capability and size of the sufficient subset are then compared with those of the entire set of speech patterns, followed by the impact on the classification performance of the influential factors.

6.4.3.1 Flat vs. tree-induced descriptors

The classification accuracies (those without temporal information) obtained by the flat and tree-induced descriptors are shown in Figures 6.5, 6.6, 6.7, and 6.8 for the four experimental datasets ITC-Irst, UPC-TALP, Freiburg-106, and ANR, respectively. From the whole set of 2,256 phone triplet categories, $\{50, 100, \ldots, 1000\}$ categories were randomly selected for the representation learning purpose and the accuracies are plotted a functions of the varying number of speech categories. Note that both the flat and tree-induced descriptors were always built upon the same subsets of phone triplets. Furthermore, it is also worth emphasizing again that once the speech-based feature extractor had been learned, it was commonly

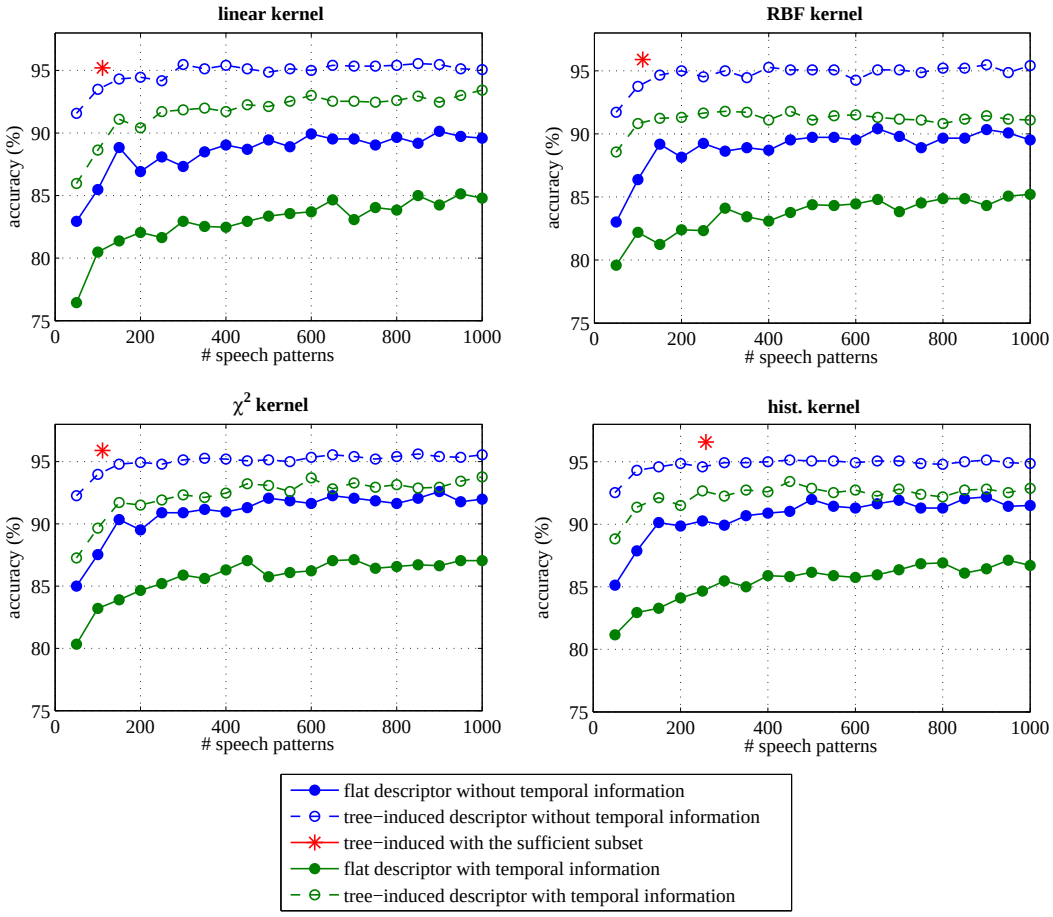

Figure 6.5: ITC-Irst dataset: Performances of the flat and tree-induced descriptors on audio event classification with different kernels.

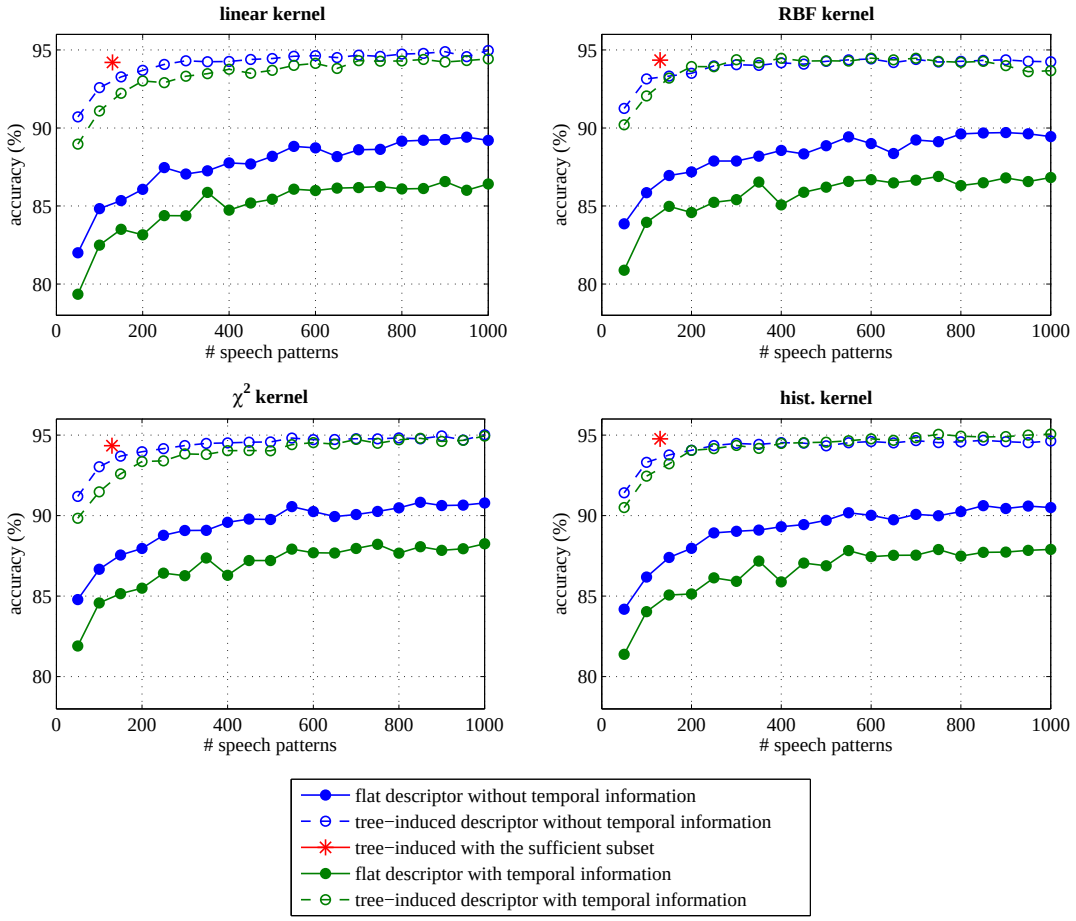

Figure 6.6: UPC-TALP dataset: Performances of the flat and tree-induced descriptors on with different kernels.

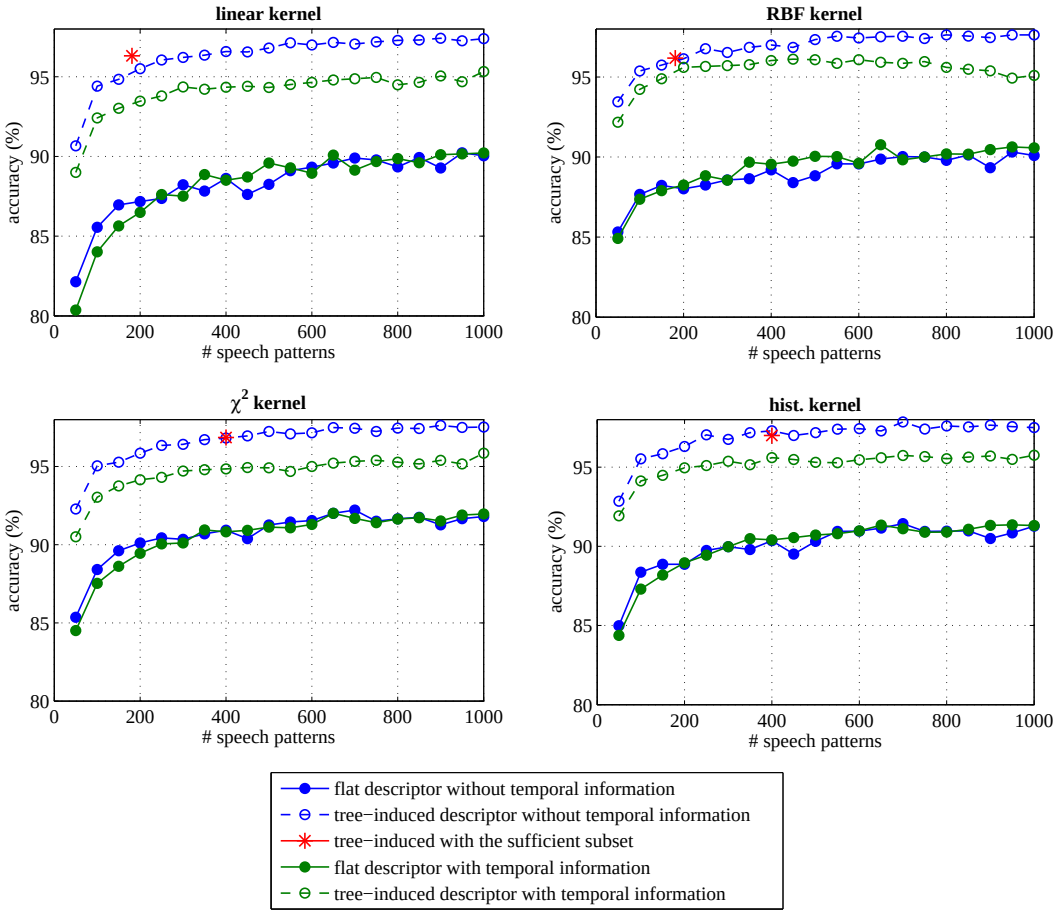

Figure 6.7: Freiburg-106 dataset: Performances of the flat and tree-induced descriptors with different kernels.

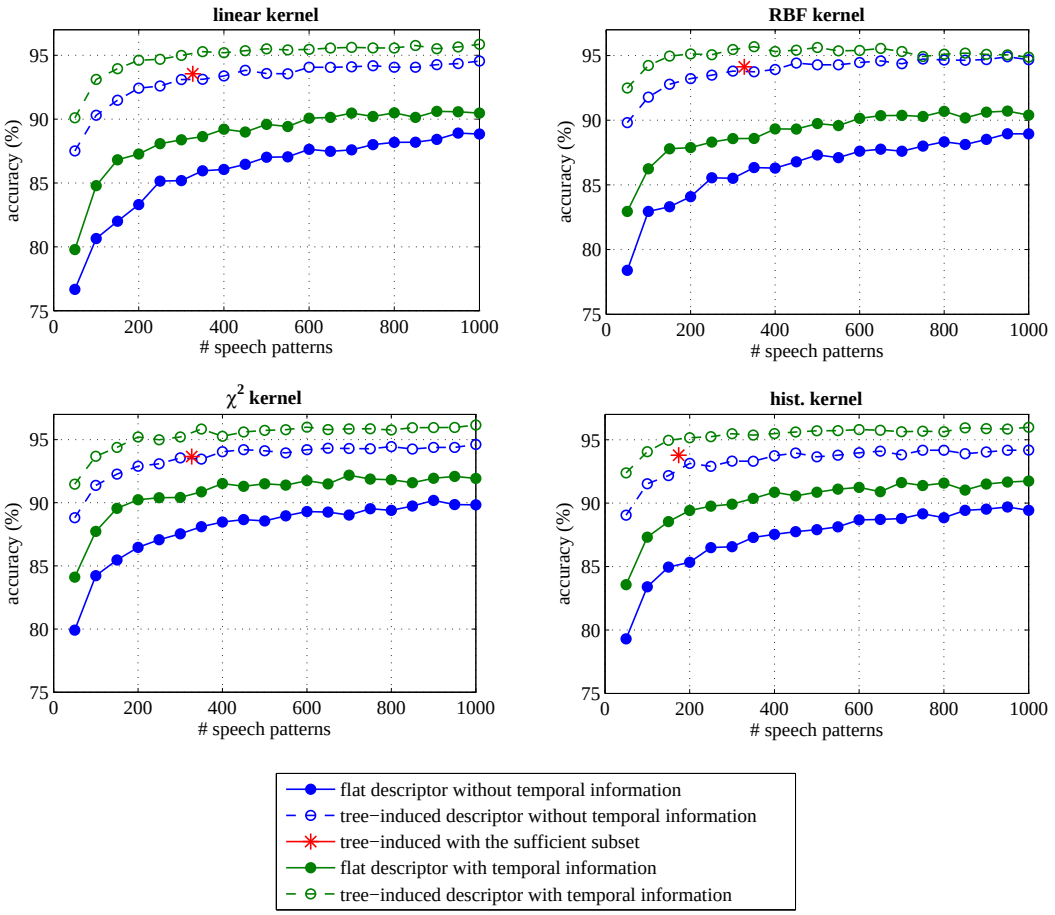

Figure 6.8: NAR dataset: Performances of the flat and tree-induced descriptors with different kernels.

Table 6.1: Average absolute accuracy gain (%) of the tree-induced descriptors over the flat descriptors.

	Linear	RBF	χ^2	Hist.
ITC-Irst	6.32	5.80	4.05	4.10
UPC-TALP	6.50	5.60	4.95	5.07
Freiburg-106	7.99	7.88	5.93	6.87
NAR	7.19	7.49	5.57	6.00

used for all four event datasets. The experiments were repeated ten times and the average accuracies are reported here.

Obviously, with the same subsets of speech patterns, the tree-induced descriptors consistently outperform the flat counterparts over all event datasets. The average absolute accuracy gains are tabulated in Table 6.1 for different kernels and event datasets. The rationale behind the performance improvement is that when the flat multi-class speech classification problem is reduced to the simple binary ones with the label tree, the meta-classes can be classified more correctly. As a result, the likelihoods with which a target audio event is classified into the meta-classes can be estimated more precisely. All of this leads to a better representation with the tree-induced descriptors compared to the flat opponents. The importance of the underlying speech classifiers will be further discussed in Section 6.4.3.5.

It can also be seen from Figures 6.5, 6.6, 6.7, and 6.8 that, more often than not, random selection of a set of speech patterns yields a good performance provided that the number of speech categories is large enough. Furthermore, the performance curves appear to saturate at some points after which adding more speech patterns results in little improvements. Last but not least, the performances of the linear classifiers are comparable with those of the nonlinear ones while they are computationally much cheaper to train and evaluate.

6.4.3.2 Sufficient subset of speech patterns vs. the whole set

In this experiment, the selection algorithm described in Section 6.3 was used to select a sufficient subset of speech patterns for representation learning. At

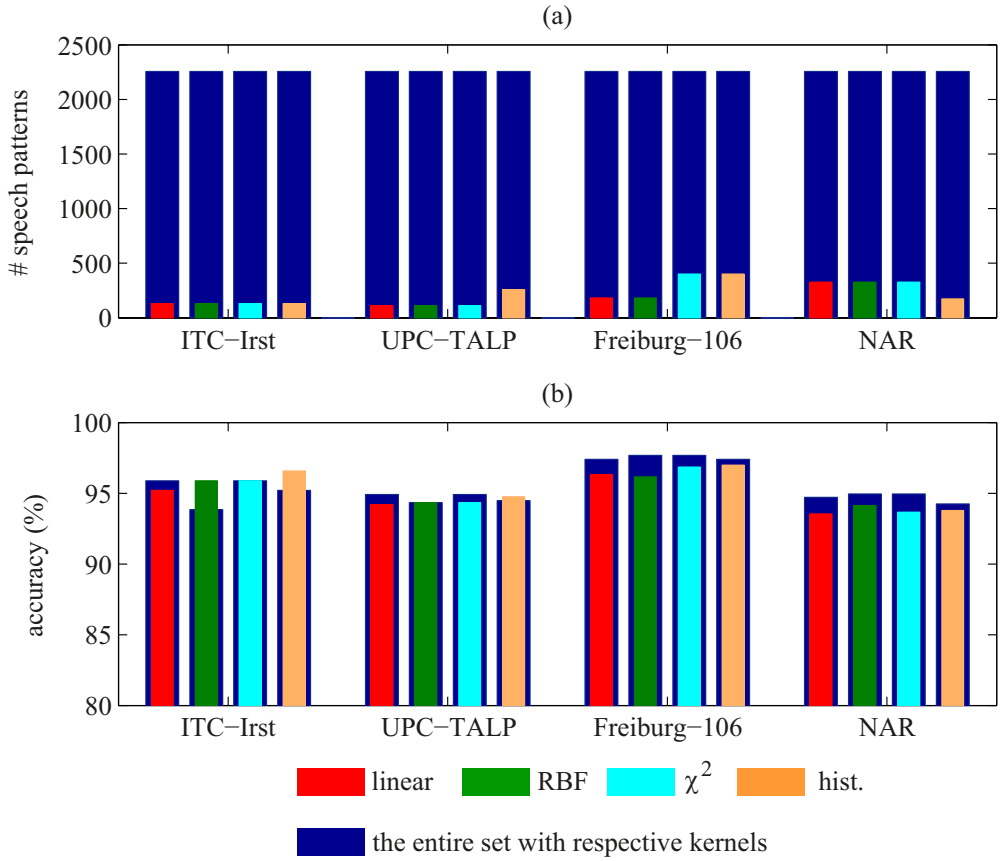

Figure 6.9: Comparison of the sufficient speech pattern subsets and the entire sets: (a) the number of speech patterns and (b) the representation capability in terms of classification accuracy.

every step of the algorithm, the next $T = 5$ speech patterns which have highest similarities to a target event category were collectively added into the current subset. Only the tree-induced descriptor, which is better than the flat one as analyzed in Section 6.4.3.1, is investigated in this experiment. Note that the selection algorithm is deterministic, therefore, the resulting subset is fixed rather than random. Moreover, these subsets are specific for different datasets and kernels.

The performances obtained by the tree-induced descriptors learned from the sufficient subsets are depicted by the red star symbol in Figures 6.5, 6.6, 6.7, and 6.8 for the four experimental datasets. It can be seen from the figures that in most of the cases these performances are above the performance curves of the random settings. Furthermore, their locations are likely in the saturation regions of the performance curves.

Table 6.2: Flat descriptors: average absolute accuracy gain (%) when the temporal information of the signals is retained.

		Linear	RBF	χ^2	Hist.
ITC-Irst		−5.6	−5.3	−5.2	−5.3
UPC-TALP		−2.6	−2.6	−2.5	−2.6
Freiburg-106		−0.1	0.4	−0.2	0.0
NAR	Overall	2.8	2.7	2.8	2.9
	Speech	8.3	8.2	7.5	7.8
	Nonspeech	−2.1	−2.1	−1.5	−1.4

In order to show that a sufficient subset can be actually representative for the entire set of the 2,256 phone triplet categories, their representation capabilities and sizes are depicted in Figure 6.9. The representation capability is defined as the accuracy of the final event classification. As can be seen from the figure, the accuracies achieved by the efficient subsets are on par to those obtainable with the entire set whereas their sizes are tremendously smaller.

6.4.3.3 Retaining temporal information

The experimental results in this section empirically reveal the effects when the temporal information of the audio signals is incorporated. In order to retain a certain degree of the temporal information, a phone triplet is divided into three constituent phones. Each phone is then decomposed into segments and described by a 53-dimensional feature vector which is the mean of per-segment features. Three feature vectors of the three individual phones are finally concatenated to make a 159-dimensional feature vector for the phone triplet. Note that the order of the constituent phones does matter here to categorize the phone triplets, therefore, the phone triplet categories in this case are different from the previous experiments. For the target audio events, as there exists no such phone components in the same way as for speech, each of them was simply divided into three equal-length segments. The feature extraction step is similar to that for speech.

Table 6.3: Tree-induced descriptors: average absolute accuracy gain (%) when the temporal information of the signals is retained.

		Linear	RBF	χ^2	Hist.
ITC-Irst		−3.1	−3.6	−2.7	−2.5
UPC-TALP		−0.7	−0.1	−0.5	0.0
Freiburg-106		−2.2	−1.5	−2.0	−1.7
NAR	Overall	1.8	1.2	2.8	2.0
	Speech	5.0	4.1	4.8	5.4
	Nonspeech	−1.0	−1.3	−0.9	−1.0

Figures 6.5, 6.6, 6.7, and 6.8 show the performance curves when the temporal information is retained in order to compare with those without the temporal information. The average absolute accuracy gains/losses are also summarized for the flat and tree-induced descriptors in Tables 6.2 and 6.3, respectively. As can be seen from the figures and the tables, the temporal information does not bring up a big advantage. It even worsens the results on the ITC-Irst, UPC-TALP and Freiburg106 datasets. The NAR dataset is an exception due to the fact that it consists of 20 speech categories out of 42 target classes (as shown in Table 5.3). It turns out that retaining the temporal information unsurprisingly benefits these speech categories but degenerates the nonspeech categories as further inspected in Table 6.2 and 6.3. Concretely, integrating the temporal dynamic of the signals does not invigorate the final representations for the target audio events, at least by the way presented in this section.

6.4.3.4 Phone triplets vs. speech words

As previously mentioned, different speech levels can be considered for speech patterns. Whereas the number of single phones is limited, phone triplets which are triple combinations of contiguous monophones were adopted for this purpose. The benefits of doing so is to create more diverging speech patterns, and hence enhance their representation capability. In addition, higher-order combinations of phones, such as words, can also be suitable for this role. The objective of this

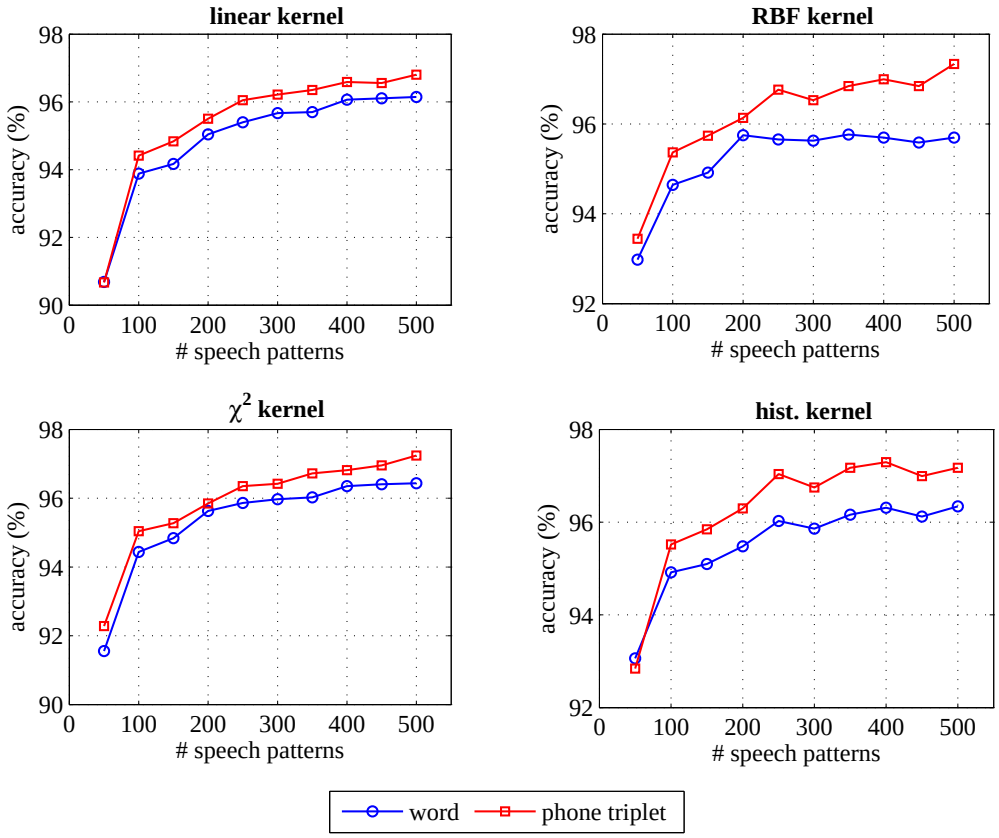

Figure 6.10: Words vs. phone triplets. The classification accuracies on the Freiburg-106 dataset with the tree-induced speech-based descriptor when words and phone triplets are used for speech patterns.

experiment is to provide a comparative study on the representation capabilities of phone triplets and speech words. The Freiburg-106 dataset was employed for this study.

Similar to the phone triplets, speech words were extracted from the TIMIT database. There are approximately 500 word categories with at least ten samples per class. This size is much smaller than the number of phone triplets since the number of constituent phones in the speech words is more than three in most of the cases. As a result, using the phone triplets offers more speech patterns for representation learning than the words. For comparison of their representation capabilities, the classification experiments on the Freiburg-106 dataset were repeated ten times for both.

Figure 6.10 shows the classification accuracies of the tree-induced descriptors when using words and phone triplets as speech patterns. As can be seen from the

figure, with the same number of speech patterns, the accuracies obtained with the phone triplets are consistently better than those obtained with the words. Specifically, the average absolute accuracy gains are 0.5%, 1.0%, 0.5%, and 0.8% with respect to the linear, RBF, χ^2, and hist. kernels, respectively. A possible explanation is that words are usually much longer than audio events and that they are special combinations of phones. As a result, audio events are often better matched with phone triplets than with words.

6.4.3.5 Importance of the underlying speech classifiers

It is studied in this section how the quality of the underlying random-forest speech classifiers influence the speech-based descriptors, and hence, the performance of the target event classification. To accomplish this, the numbers of trees in the random-forest classifiers in Sections 6.2.1 and 6.2.2 were varied in the range $\{25, 50, \ldots, 200\}$ and their out-of-bag (OOB) errors were recorded. The OOB errors are estimated internally during the forest construction [18]. In general, increasing the number of trees is expected to produce a performance gain at the cost of increasing computation.

Figure 6.11 shows the OOB errors of the random-forest speech classifiers for both the flat and tree-induced cases. In particular, for the latter the OOB error averaged over the binary classifiers associated with a label tree is reported. As can be seen from the figure, the error curves exhibit similar patterns for both cases. As expected, the error curves escalate as the speech classification problem becomes more complex with the increasing number of speech patterns. In addition, they expose a downward shifting trend, i.e. better performance, as the number of trees increases. However, the improvement decreases incrementally with an increasing number of the trees. When the number of trees is large enough, e.g. at 150 trees, adding more trees leads to insignificant performance gains. Note that the error scales in the tree-induced case are significantly smaller than those of the flat case. This can be explained by that the multi-class classification problem in the flat case has been reduced into multiple simpler and easier binary ones in the tree-based case. This reflects the superiority of the tree-induced descriptors over the flat counterparts in the final classification task.

The random-forest speech classifiers with different number of trees were then employed during learning the speech-based descriptors for the target audio events. The event classification accuracies (those obtained with the χ^2 kernel) are shown

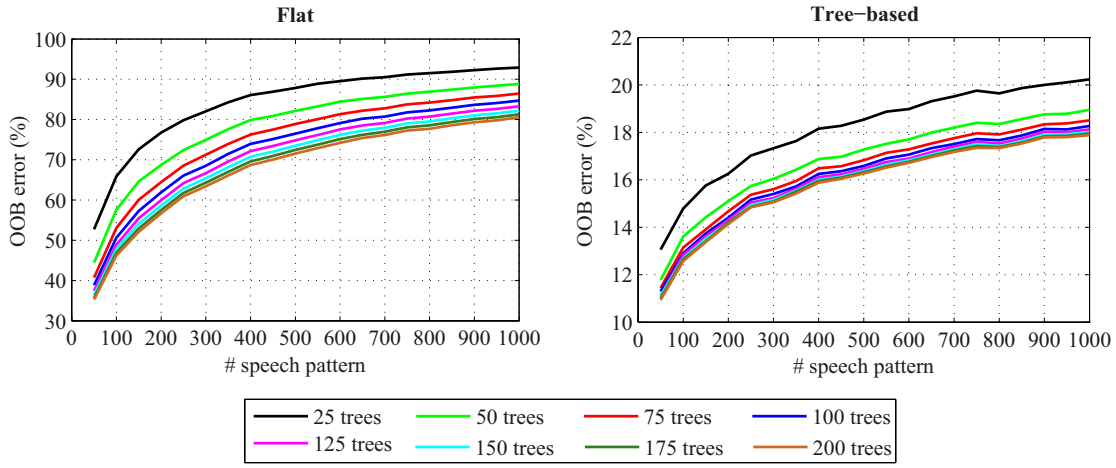

Figure 6.11: The average OOB errors of the random-forest speech classifiers as functions of the number of trees.

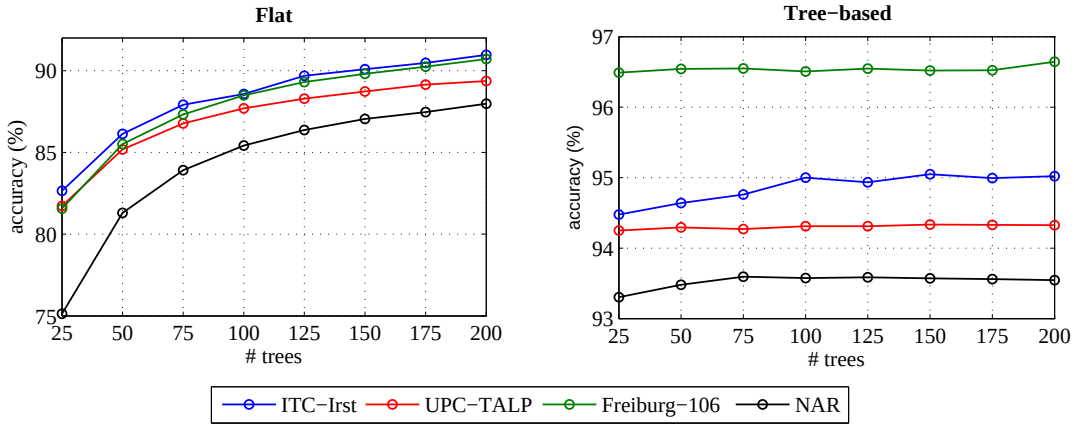

Figure 6.12: Average event classification accuracies as functions of the number of trees of the random-forest speech classifiers.

in Figure 6.12 as functions of the number of trees. For clarity, at each number of trees, the average performances over different numbers of speech categories in $\{50, 100, \ldots, 1000\}$ are reported. Two different patterns can be seen from the figure for the flat and tree-induced cases. For the former, increasing the number of trees obviously leads to performance gains although they are gradually diminishing with the increase of the number of trees. In contrast, the performance curves in the latter case are almost flat, indicating very small variations in the classification accuracies. These results imply that the number of trees of the speech classifiers is more important for the flat descriptors than for the tree-induced ones. This

is reasonable since the multi-class classification problems in the flat case need a strong classifier, i.e. a large number of trees, while the simple binary classification problems in the tree-based case can be easily coped with simpler classifiers.

6.4.3.6 Using speech-based descriptors as additional features

The objective of this experiment is to investigate how the presented generic speech-based descriptors can be used to improve another classification system with some fusion schemes when they are treated as additional features. The tree-induced descriptors derived from the sufficient subsets with respect to the χ^2 kernel were employed to be integrated with those obtained by the baselines **BoW**, **PBoW-2**, **PBoW-3**, and **PBoW-4** in Section 5.3.3. The results with different codebook sizes of the baseline systems are then analyzed. Different descriptors were combined using the extended Gaussian kernel given in Eq. (5.12) and the classification was accomplished using one-vs-one SVMs.

The fusion results are illustrated in Figure 6.13. It is clearly shown for both Freiburg-106 and NAR that the classification performances are significantly improved. For the former, the gains of $> 2\%$ absolute (averaged over different codebook sizes) are seen for all the baselines whereas those gains for the latter are $> 1\%$ absolute. In case of ITC-Irst and UPC-TALP, however, the improvements appear inconsistent. Performance drops can also be seen especially at a large codebook size, i.e. 250. Nevertheless, the overall results are positive as shown in Table 6.4, which summarizes the accuracy gains averaged over different codebook sizes.

6.4.4 Performance Comparison

This section provides an overall performance comparison of the presented speech-based systems, the baselines, their fusion systems, and the best reported results on the experimental datasets. The performances of the speech-based systems are reported using those obtained with the tree-induced descriptors learned from the sufficient subsets. For the baselines **BoW**, **PBoW-2**, **PBoW-3**, and **PBoW-4**, their best performances amongst different codebook sizes and kernels were used for comparison (cf. Section 5.3.3). The fusion systems were implemented by integrating the speech-based descriptors with the corresponding best baseline systems using the fusion scheme in Section 6.4.3.6. Finally, the best results on the ITC-Irst,

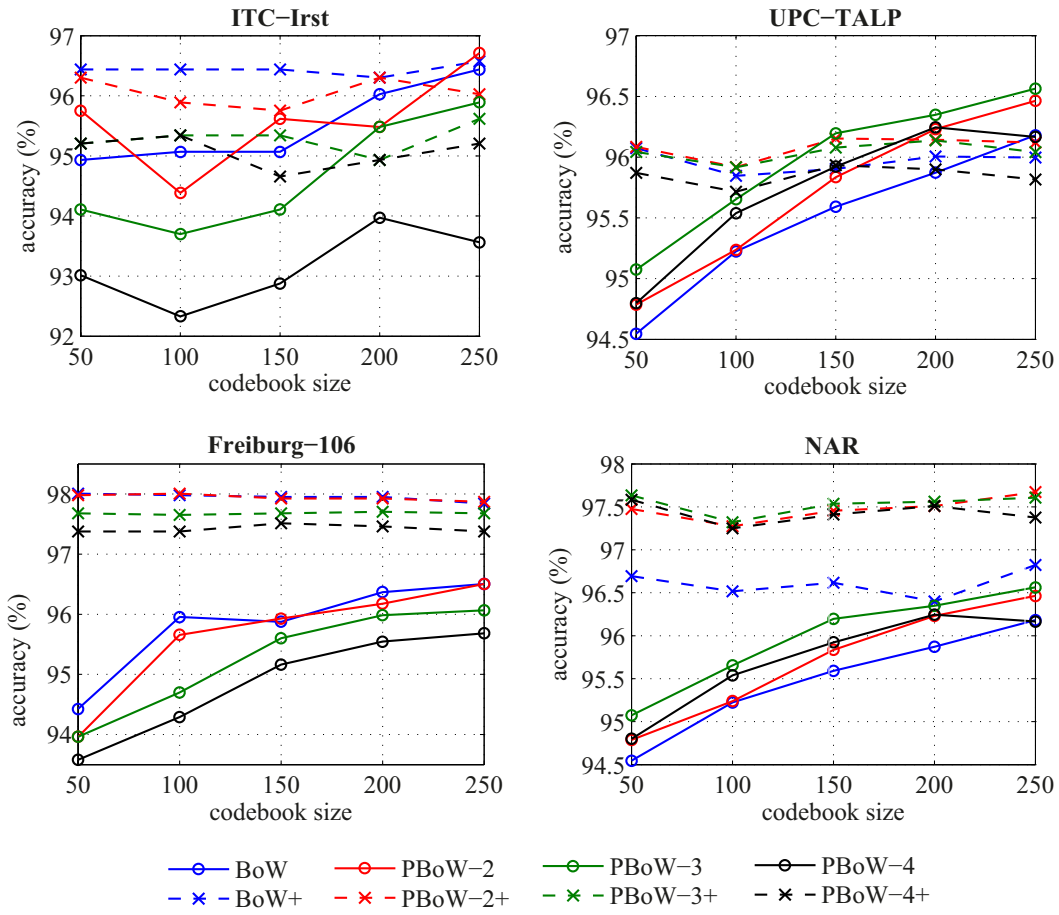

Figure 6.13: Performances of the fusion systems (i.e. the combinations of the baselines and the speech-based systems) compared to those of the standalone baseline systems. Note that the fusion systems are denoted with the additional '+' symbol.

Table 6.4: Average absolute accuracy gains (%) obtained by the fusion systems (i.e. the combinations of the baselines and the speech-based descriptors) over the standalone baseline systems.

	BoW	**PBoW-2**	**PBoW-3**	**PBoW-4**
ITC-Irst	0.9	0.5	0.6	1.9
UPC-TALP	0.5	0.4	0.1	0.1
Freiburg-106	2.1	2.3	2.4	2.6
NAR	1.1	1.8	1.6	1.7

UPC-TALP, Freiburg-106, and NAR datasets were reported in the works of Temko et al. [214], Butko et al. [22], Hertel et al. [97], and Maxime et al. [139], respectively. The performance comparison is shown in Table 6.5. Note that, to agree with the results in previous works [97, 205], the performances on the Freiburg-106 dataset were reported in terms of F1-score instead of accuracy.

It can be seen from the table that, overall, good classification accuracies are obtained with the speech-based descriptors alone. They are on par to that of the best baseline (i.e. PBoW-2) on the ITC-Irst dataset and even better than the best reported accuracy in [214] on this dataset. For the Freiburg-106 dataset, although the speech-based accuracy is inferior to that obtained with deep CNNs in [97], it outperforms those of all the baselines. On the other hand, the speech-based accuracies on the UPC-TALP and NAR datasets are not comparable to those of the baselines. However, it should be emphasized that in contrast to the baselines, the features from the audio events themselves have not been taken into account when classifying with the speech-based descriptors alone. Integration of both sources (i.e. the speech-based descriptors and the baselines' descriptors) is particularly efficient for the Freiburg-106 and NAR datasets. It leads to significant accuracy gains and reduces the gap to the strong deep CNNs in [97] by half on the Freiburg-106 dataset. More encouragingly, the fusion system sets the best performance, outperforming all the competitors on the NAR dataset. Last but not least, although the speech-based descriptors are not as capable as the audio phrases and bank-of-regressors presented in Chapter 5, they offer the unique advantage of being generic.

Table 6.5: Performance comparison (%) between the speech-based systems, the baseline systems, their fusion systems, and the best results reported in previous works.

	Speech-based	Baseline				Speech-based + Baseline				Best reported
		BoW	**PBoW-2**	**PBoW-3**	**PBoW-4**	**BoW+**	**PBoW-2+**	**PBoW-3+**	**PBoW-4+**	
ITC-Irst	96.6	96.4	96.7	95.9	94.0	96.6	96.0	95.6	94.9	95.6
UPC-TALP	94.8	96.3	96.5	96.6	96.2	96.0	96.1	96.0	95.9	92.9
Freiburg-106	96.8	95.9	96.0	95.8	95.2	97.6	97.6	97.4	97.1	98.3
NAR	94.1	94.5	95.8	96.4	96.1	96.8	97.5	97.6	97.4	97.0

7 AED Revisited: False Positive Reduction

In this chapter, the audio event detection task will be revisited to study one of its important aspect: false positive reduction. Fundamental differences between audio event detection and classification will be analyzed. Afterwards, these dissimilarities will be leveraged for an improved generic detection pipeline that supports false positive reduction. In this pipeline, a detection system is appended by a verification step for augmentation purposes, in which a high-quality classifier is employed to postprocess event hypotheses outputted by the detection system and reject false alarms.

7.1 False Positive Reduction in AED

So far, the majority of research have focused on improving the overall detection performance in terms of accuracy and little attention has been paid to the important aspect of false positive reduction. False positives, i.e. event instances that are spuriously detected by a detection system, and subsequently draw attention to them, are arguably one of the most important problems faced by different applications like ambient intelligence and surveillance. To the best knowledge of the author, this is the first work explicitly addressing this problem. The goal of false positive reduction is obviously achievable by improving the overall performance towards a perfect system which makes no mistakes. Hence, it was implicitly addressed in many works since the task was introduced. However, reaching such a perfect detection system is hard, if not impossible. The problem faced here is different from prior works in essence. The objective is to reduce the false positives of detection systems with less effort given their state-of-the-art performance which is far from perfect in practice.

7.2 Audio Event Detection vs. Classification

There is a common observation that audio event classification is easier to deal with than detection. For example, on the ITC-Irst dataset [214], the best accuracies of 98.4% (cf. Table 5.9(a)) and 93.6% (even with multiple channel fusion, cf. Table 4.1) are obtained for the classification and detection tasks, respectively. This observation is also well-known in the CLEAR 2006 challenge [214]. This, however, has been accepted as a fact so far and a careful analysis is lacking. This can be explained by the fundamental differences between the classification and detection tasks. The reason is two-fold. First and more obviously, the detection task needs to discriminate not only the event categories of interest (as in the classification task) but also the target event categories as a whole from highly rich background sounds. Second, for the classification task, one has access to the global context of the events. On the contrary, in the detection task, boundaries of the events are not known in advance and one usually needs to rely on unreliable local audio features for inference.

Furthermore, these fundamental dissimilarities result in a pitfall of event detectors which are based on the common detection-by-classification scheme, such as those in [28, 85, 120, 180]. These detectors attempt to build strong event classification models and subsequently employ them to detect events in continuous streams with a sliding window. Since the classifiers are trained on complete events, they expect to be presented with complete events to guarantee a good performance. However, due to high intra- and inter-class temporal variations of audio events, it is almost infeasible to choose a good-for-all window length that exactly captures complete events. This causes a mismatch between training and testing data, which subsequently deteriorates the accuracy of the classification models and the accuracy of detection systems. Although one can circumvent this issue by training classifiers on equal-sized segments of the events, such as those in [195, 212, 214], the problem remains unsolved. By dividing the events into equal-sized segments, one has increased the complexity of the data distribution. This makes the classification problem harder to solve than the original one considering the entire events as training examples. All of this, again, results in the degeneration of the classification models.

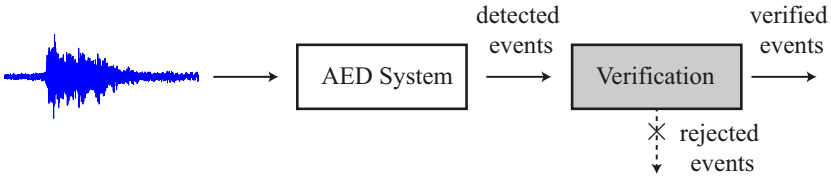

Figure 7.1: Audio event detection with verification.

7.3 Improved Detection Pipeline with Verification

The improved detection pipeline proposed here is inspired by investigating the performance gap between a detection system on continuous streams and a classification system on isolated events as discussed above. The idea is to augment a detection system with a verification step where a high-quality classification model will be employed to verify the detected event hypotheses as in Figure 7.1. At this verification step, an event hypothesis with a class label c_{detected} outputted by the detection system will be rejected when it is classified with a mismatched label $c_{\text{classified}} \neq c_{\text{detected}}$ by the classifier. Eventually, instead of making hard decisions early on, the false positive hypotheses outputted by a detection system will be rejected by the verification step. As a result, the detection precision will be enhanced, leading to improvements in the overall detection performance. Unlike the common detection-by-classification scheme [85, 180, 208, 212], this can be considered as a novel scheme to utilize a trained event classifier for the detection task.

The rationale behind the improved pipeline is motivated from the differences between the classification and detection tasks as discussed above. Since the detection task relies on local features, the wrongly detected events are usually difficult ones whose local features are not reliable and cause wrong detections. However, after the detection step, one has obtained the estimated boundary of the detected events and therefore, has access to their more or less global contexts. As a consequence, the mismatch between training and testing data is mitigated and an event classifier is expected to perform well.

7.4 Experiments on the ITC-Irst Dataset

The experiment in this section studies how the verification step in the improved detection pipeline benefits detection performance on the ITC-Irst dataset [214].

Three detection systems presented in Section 3.3 of Chapter 3, namely the regression forest based detector (Weighting Scheme 3) and the two baseline detectors **SVM** and **HMM** were employed as base detectors. They were coupled with different classifiers that have been developed in Chapters 5 and 6, including the best baseline classifiers (**BoW**, **PBoW-2**, **PBoW-3**, and **PBoW-4**), the BoP-based classifiers (hard **BoP-1**, hard **BoP-2**, hard **BoP-3**, soft **BoP-1**, soft **BoP-2**, and soft **BoP-3**), the BoR-based classifiers (**BoR-MV**, **BoR**, and **BoR+**), and the speech-based classifier. The classifiers play the role of the verifier in the proposed detection pipeline. It should be noted that the speech-based classifier employed here is the tree-induced one learned from the sufficient subset of speech patterns.

The overall detection results with the verification step are shown in Table 7.1 for all possible detector-classifier combinations. In general, the verification enhances the precision values of the detection systems at the cost of decreasing the recall values. However, the recall drops are smaller than the precision gains, which lead to overall F1-score improvements. On average, the absolute F1-score improvements are 6.7%, 1.3%, and 0.9% for the **SVM**, **HMM**, and regression forest detectors, respectively. These results demonstrate the effectiveness of the proposed detection pipeline.

As expected, the detected hypotheses are not perfectly segmented and most likely contain certain segmentation errors. On the other hand, the classifiers were trained on manually annotated examples, i.e. perfect segmentation. This results in a certain degree of mismatch between training and test data. The results with verification in Table 7.1 also reveal the robustness of the classifiers to the data mismatch, indicated by the recall losses. The larger a reduction is, the more likely true positive hypotheses are accidentally rejected by a verifier. Overall, the BoP-based and BoR-based classifiers (except for **BoR-MV**) are more robust than both the baselines and the speech-based classifiers. In particular, for the case of the **SVM** detector, the **PBoW-2**, soft **BoP-1**, soft **BoP-2**, and speech-based classifiers appear to be the most robust ones as no true positives are rejected by them. However, the speech-based classifiers turn out to be too aggressive on the **HMM** and regression forest detectors, resulting in largest recall reductions.

These results also help us to gain insight into the behaviors of different detection systems. The **SVM** system tends to retain a lot of event hypotheses which are explained by its high recall (87.7%) and low precision (80.1%). In contrast, the **HMM** system conservatively retains a relatively small number of hypotheses

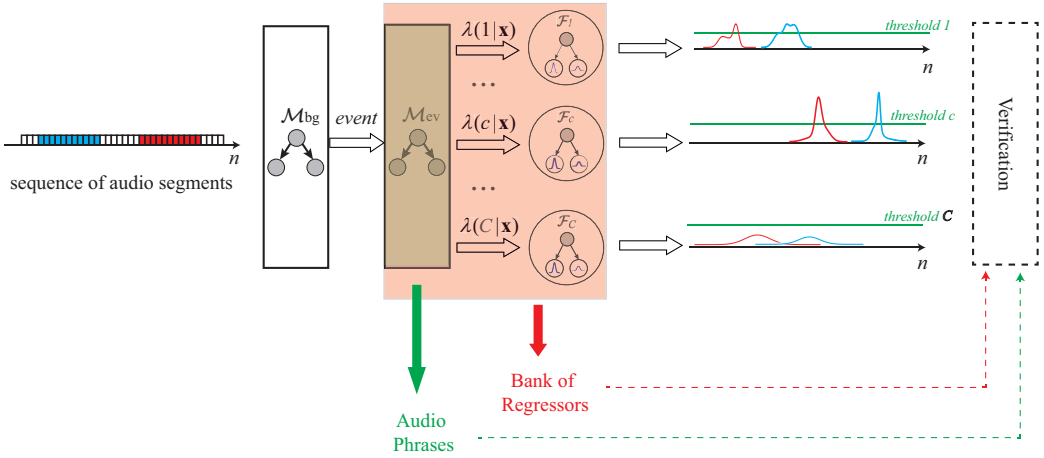

Figure 7.2: The BoP and BoR features, extracted from the same ingredients of the regression forest detector.

indicated by its high precision (87.4%) but low recall (81.5%). Consequently, the verification step is able to reject a lot more false positive hypotheses of the **SVM** system to significantly boost the precision by 13.8% absolute (averaged over all verifiers) whereas the precision gain of the **HMM** system is more subtle, 5.2% absolute on average. Interestingly, although the regression forest detector obtains much higher precision and recall than the other two (i.e. the detected hypotheses are of high fidelity), the verification step can still further help to reject false alarms and yield an average precision gain of 3.1% absolute.

Last but not least, although all detector-classifier combinations yield consistent improvements on the overall detection performance, it should be more favorable to use regression forest detection in combination with the BoP-based or BoR-based classifiers. This is partly because of the state-of-the-art results (as in Table 7.1, 94.7% on F1-score is achievable with the combination of the regression forest detector and the soft **BoP-2** classifier) and partly because the in situ BoP and BoR features are extracted using the available components of the detector as illustrated in Figure 7.2. It gets rid of the necessity for building additional components for other classifiers.

Table 7.1: Overall detection results of different detection systems with and without verification. For those with verification, the absolute performance gains and losses compared to those without verification are highlighted in blue and red, respectively.

		SVM			**HMM**			**Regression**		
		F1-score	prec.	recall	F1-score	prec.	recall	F1-score	prec.	recall
w/o verification		83.7	80.1	87.7	84.4	87.4	81.5	93.1	93.6	92.7
w/ verification	**BoW**	90.3 ↑ 6.6	93.4 ↑ 13.3	87.4 ↓ 0.3	85.9 ↑ 1.5	94.5 ↑ 6.1	78.8 ↓ 2.7	93.4 ↑ 0.3	96.5 ↑ 2.9	90.4 ↓ 2.3
	PBoW-2	90.5 ↑ 6.8	93.6 ↑ 13.5	87.7 ↓ 0.0	86.8 ↑ 2.4	94.8 ↑ 7.4	80.0 ↓ 1.5	93.7 ↑ 0.6	96.3 ↑ 2.7	91.2 ↓ 1.5
	PBoW-3	89.7 ↑ 6.0	92.4 ↑ 12.3	87.3 ↓ 0.4	85.7 ↑ 1.3	94.5 ↑ 7.1	78.4 ↓ 3.1	93.6 ↑ 0.5	96.1 ↑ 2.5	91.2 ↓ 1.5
	PBoW-4	89.7 ↑ 6.0	92.2 ↑ 12.1	87.4 ↓ 0.3	85.0 ↑ 0.6	94.3 ↑ 6.9	77.4 ↓ 4.1	93.6 ↑ 0.5	96.3 ↑ 2.7	91.1 ↓ 1.6
	hard **BoP-1**	90.7 ↑ 7.0	94.7 ↑ 14.6	87.0 ↓ 0.7	86.2 ↑ 1.8	92.0 ↑ 4.6	81.1 ↓ 0.4	94.4 ↑ 1.3	97.1 ↑ 3.5	91.9 ↓ 0.8
	hard **BoP-2**	90.8 ↑ 7.1	94.7 ↑ 14.6	87.1 ↓ 0.6	85.8 ↑ 1.4	91.8 ↑ 4.4	80.5 ↓ 1.0	94.7 ↑ 1.6	97.1 ↑ 3.5	92.3 ↓ 0.4
	hard **BoP-3**	90.6 ↑ 6.9	94.6 ↑ 14.5	87.0 ↓ 0.7	85.2 ↑ 0.8	90.8 ↑ 3.4	80.3 ↓ 1.2	94.3 ↑ 1.2	97.1 ↑ 3.5	91.6 ↓ 1.1
	soft **BoP-1**	91.0 ↑ 7.3	94.6 ↑ 14.5	87.7 ↓ 0.0	86.3 ↑ 1.9	92.2 ↑ 4.8	81.1 ↓ 0.4	94.4 ↑ 1.3	96.9 ↑ 3.3	92.1 ↓ 0.6
	soft **BoP-2**	90.9 ↑ 7.2	94.3 ↑ 14.2	87.7 ↓ 0.0	85.9 ↑ 1.5	91.7 ↑ 4.3	80.8 ↓ 0.7	94.1 ↑ 1.0	96.6 ↑ 3.0	91.8 ↓ 0.9
	soft **BoP-3**	90.7 ↑ 7.0	94.2 ↑ 14.1	87.5 ↓ 0.2	85.9 ↑ 1.4	91.6 ↑ 4.2	80.8 ↓ 0.7	94.4 ↑ 1.3	97.0 ↑ 3.4	91.9 ↓ 0.8
	BoR-MV	88.3 ↑ 4.6	92.6 ↑ 12.5	84.4 ↓ 3.3	84.3 ↑ 0.1	92.1 ↑ 4.7	77.8 ↓ 3.7	92.9 ↑ 0.2	96.1 ↑ 2.5	89.9 ↓ 2.8
	BoR	91.0 ↑ 7.3	94.7 ↑ 14.6	87.5 ↓ 0.2	85.9 ↑ 1.5	92.6 ↑ 5.2	80.0 ↓ 1.5	94.2 ↑ 1.1	96.6 ↑ 3.0	91.9 ↓ 0.8
	BoR+	91.0 ↑ 7.3	95.1 ↑ 15.0	87.3 ↓ 0.4	85.9 ↑ 1.5	92.6 ↑ 5.2	80.0 ↓ 1.5	94.2 ↑ 1.1	96.6 ↑ 3.0	91.9 ↓ 0.8
	Speech-based	90.5 ↑ 6.8	93.6 ↑ 13.5	87.7 ↓ 0.0	83.9 ↑ 0.5	92.5 ↑ 5.1	76.7 ↓ 4.8	93.4 ↑ 0.3	97.5 ↑ 3.9	89.6 ↓ 3.1

8 Conclusion and Future Works

Concretely, this thesis has focused on dealing with both audio event detection and classification tasks. Due to the fact that many audio events exhibit strong temporal structures, the main idea is to model them to leverage detection and classification. This is not an easy task since temporal structures of audio events are much more complex than, for example, those of speech. This limits the capability of both detection-by-classification and ASR-based approaches for detection purpose since they are either unable or inefficient in capturing long temporal tendencies of event signals. With this in mind, a new approach has been proposed in this thesis for the detection task. The idea is to model relative positions of the audio segments, into which event instances are decomposed, to the event onsets and offsets using a random regression forest model. The empirical results on the ITC-Irst dataset show that the detection system based on this approach outperforms the baselines based on the common approaches by a large margin. Furthermore, the proposed system comprises an intrinsic mechanism for early event detection with reliability and allows multi-channel fusion to be carried out in a simple additive manner. Finally, a generic improved detection pipeline has further been introduced, augmenting a certain detection system with a verification step where a high-quality classifier is employed to verify and reject detected false positives.

Turning to the classification task, two data-specific representations have been presented, namely bag-of-phrases and bank-of-regressors. These learned features are capable of encoding temporal configurations of audio events, explaining their state-of-the-art performance on four different datasets. To alleviate the need for generic descriptors for audio events, one such descriptor has been introduced in this thesis using similarities between a target event instance to different speech phone triplets. While they demonstrate good classification accuracy, they can also serve as a valuable external source to improve an existing classification system. When playing the role of the verifier in the improved detection pipeline mentioned above, the classifiers trained on these representations gain significant improvements in the detection performance.

8.1 Contributions

Random regression forests AED system. For each event category of interest, the annotated event instances in training data were decomposed into multiple overlapping audio segments. A class-specific random regression forest was then trained using the set of audio segments to model the relative positions of the segments to the event onsets and offsets. During testing, via the regression model, an unseen audio segment gave estimations where the onset and offset of the target event are most likely in a continuous signal. The individual estimates made by all segments were integrated to form the detection confidence scores. Via thresholding, the event onset and offset positions were eventually determined as temporal positions at local maxima of the confidence scores. Furthermore, in order to weight the contributions of individual segment-wise estimates into the final confidence scores, three different weighting schemes have also been studied using the probabilities of a segment-wise event classifier. The experimental results on the ITC-Irst dataset demonstrated the superiority of the proposed system over two baselines which rely on the common detection-by-classification and ASR-based approaches. Compared to the best baseline system, an absolute gain of 9.4% in terms of F1-score was obtained while the detection error rate was lowered by 15.7% absolute.

Intrinsic early detection ability. The detection function of the proposed regression forest detection system was proved to fulfill the monotonicity requirement, enabling early detection with reliability. That is, a target event instance that is detectable by the system can be detected early in time before it finishes without losing any overall detection performance. This finding was demonstrated for all target event categories of the ITC-Irst dataset.

Additive multi-channel fusion framework. Using the regression forest detectors as basic components, a simple yet efficient multi-channel fusion framework was introduced to leverage the spatial information of distributed microphones. For each individual microphone, a channel-specific detector was trained. The fusion system then additively accumulated the onset and offset estimation confidence scores made by the channel-wise detectors to obtain the fused confidence scores. The experiments on the ITC-Irst dataset with five selected microphones showed that an average absolute gain of 1.5% was achieved by the fusion system compared

to the channel-wise components. Furthermore, the improvements on F1-score are monotonic when the available microphones were integrated into the fusion system.

Audio phrases and BoP representation. In order to overcome the limitations of the popular BoW model, the concept of audio phrases has been proposed where an audio phrase is the combination of multiple audio words. Afterwards, the BoP representation was derived in a similar manner to that of the BoW model. To alleviate the issue of high dimensionality resulted by high-order audio phrases, a method to learn a discriminative compact codebook, in which a segment-wise classifier was employed for codebook matching, was further presented. The BoP representation demonstrated good performance on four different datasets, outrunning all the baselines and previously reported results and setting state-of-the-art performance on the UPC-TALP and Freiburg-106 datasets.

BoR representation. Using the class-wise regression forests used in the detection task, a compact yet discriminative representation was derived by stacking the regressors in a bank for feature extraction. Each entry of the learned feature vector is the response of the corresponding regressor on the input event, which can be interpreted as how good the input instance aligns with the temporal structure modeled by the regressor. Stacking multiple regressors allows the shared features between different target event classes to be encoded. Similar to the BoP, the BoR representation outperformed all the baselines and previously reported results on all four experimental datasets and achieved state-of-the-art performance on the ITC-Irst and NAR datasets.

Speech-based generic representations. To remedy the need for generic representations for audio events, one such representation has been introduced. Utilizing a set of speech patterns (i.e. phone triplets) as basic acoustic concepts, a target event instance is represented by its similarities to these speech patterns which are obtained via a simple multi-class speech classifier or a label-tree based hierarchy of binary classifiers. While this representation alone showed good classification accuracies on the experimental datasets, they can also serve as a valuable external source to improve an existing system as demonstrated for the BoW models and the pyramid BoW models.

Generic improved detection pipeline. On revisiting the detection task, this improved detection pipeline was proposed to actively address false positives which are unavoidably outputted by a detection system. This was accomplished by appending a verification step to the detection pipeline at which a high-quality

classifier is employed to verify and reject the false alarms. Exhaustively coupling all the detectors and the classifiers introduced in this thesis in the proposed pipeline demonstrated consistent improvements on overall detection performance in terms of F1-score.

8.2 Future Works

The community has recently made efforts to enhance audio event classification/detection in more severe scenarios. In addition, inspired by the success of recent advances of machine learning in many fields, there are currently several moves to utilize them to leverage the development of the field. The work in this thesis can be further developed in different directions.

Foremost, since audio events have durations, event overlap arises when two or more audio events have their durations partly or completely overlapping each other, i.e. they occur simultaneously in time. It has been shown in recent DCASE challenges [1, 208] that handling overlapping events is very challenging. One approach could be to entangle their mixed frequency contents and to treat the problem as a feature selection one. Intuitively, when an event instance is partly overlapped by others, it is still recognizable using its nonoverlapped parts. In the harder scenario of being fully overlapped, one could probably rely on local features, local frequency-temporal patches for example, to identify the target event. In both cases, the system would need to know both positive and negative examples to be able to select discriminative features to tell apart the positives from the negative ones. While the regression forest detectors proposed in this thesis are superior for nonoverlapping event detection, they are not supposed to handle event overlaps well. The class-wise regression forests were trained on positive examples only and, therefore, know nothing about negative ones. In order to overcome this, they need to be equipped with a feature selection capability. One possibility is to train the forests jointly for classification and regression. For instance, a split node would, alternatively or randomly, perform data splitting to optimize both classification and regression objective functions as in [198, 199]. The classification-oriented training forces a forest to select discriminative features from polyphonic mixtures to separate positive examples from the rest. The regression-oriented training then follows up to model temporal structures of audio events as in regular regression

forests. Using this strategy, the author's submission to the most recent DCASE 2016 challenge showed promising results on overlapping event detection [176].

The systems presented in this work were evaluated on different indoor datasets with more or less controlled conditions (e.g. good-quality microphones and low noise levels). Future work in the field would consider in-the-wild tasks to understand natural audio which has different characteristics on reverberation, echo, and overlap. It would be interesting to analyze the systems' behaviors on more general conditions. This would allow one to evaluate the robustness of the proposed system.

The community is currently in need for open-source large corpora which are inspired by public datasets in other fields, such as ImageNet [45] with nearly 14.2 million static images in computer vision or Librispeech [159] with thousands of hours of speech in ASR. Such large datasets would be extremely important for the development of the field and even inevitable for the applications like video/multimedia analysis [9, 96, 217]. Although a few recent works reported results on large datasets, such as SoundNet [9] and Youtube-100M [96], these datasets are not public. There is currently a running project to resolve this open-data bottleneck, namely AudioNet [8]. This project aims at creating a corpus of audio annotations on the Yahoo Flickr Creative Commons 100 Million (YFCC100M) dataset [217] and making it publicly free for research use. When available, such a dataset would enable studying the tasks in a much larger scale. On the other hand, the algorithms employed in this thesis should work well on the currently available datasets, such as those used in the experiments, which are relatively small. However, they are not supposed to handle such massive amount of data. It is partly because the number of trees would soar exponentially and partly because the number of class-wise regression components would also escalate with the number of target event categories. It is necessary to seek for alternative algorithms that are scalable with the large amount of data. Deep learning algorithms [82, 124] appear to be perfect candidates. More specifically, decision forest classifiers and regressors in the pipeline in Figure 3.6 can be simply replaced by DNNs or CNNs trained for classification and regression tasks, respectively. Furthermore, a multitask network as in [188] could even handle both classification and regression tasks jointly. One out of many advantages of these algorithms is that they can incrementally learn from small batches of data to yield good suboptimal models at the end. So far, they have been underused due to the lack of sufficiently large data. With large

datasets made available in the future, one can expect breakthroughs as recently seen in many other fields.

Another possible utilization of large datasets like AudioNet [8] is to explore them to benefit a specific task at hand. A possibility is to learn useful generic features from a large amount of data and then directly use or fine-tune them for different lower-scale tasks, similarly to [9] in acoustic scene classification or [219] in video analysis. Unfortunately, comprehensive annotation for such amount of data would require tremendous efforts over several years. Supervised feature learning approaches like [219] would not be possible until then. A possible way to go around this issue is to rely on unsupervised feature learning on unannotated data which recently showed promising results on different computer vision tasks [100, 203, 232]. Another possibility is to utilize the available annotations for videos to constrain learning features for audio in a transfer learning setting as in [9]. In addition, as shown in Chapter 6, one could train a generic feature extractor for audio events using speech data. The fact is that there exist more than 6,900 languages in the world [83] and many annotated corpora are available beside TIMIT [64], such as SWITCHBOARD [79], Wall Street Journal [164], GlobalPhone [200], Librispeech [159] to mention a few. This opens up enormous opportunities to explore for learning representations from speech. Using different levels and different languages would result in different representations. Their combinations would offer even more opportunities. Furthermore, speech-based generic features would not be limited to audio event representation, they could be applied to any other variants of audio signals such as music and even speech. At least, the induced descriptors can act as additional sources to improve performance of existing systems.

Bibliography

[1] http://www.cs.tut.fi/sgn/arg/dcase2016/.

[2] S. Adavanne et al. *Sound Event Detection in Multichannel Audio Using Spatial and Harmonic Features*. Tech. rep. Detection, Classification of Acoustic Scenes, and Events 2016, 2016.

[3] F. Alias, J. C. Socoro, and X. Sevillano. "A Review of Physical and Perceptual Feature Extraction Techniques for Speech, Music and Environmental Sounds". In: *Applied Sciences* 6.5 (2016), p. 143.

[4] D. Amodei et al. "Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin". In: *Proc. 33rd International Conference on Machine Learning (ICML)*. 2016, pp. 173–182.

[5] K. Ashraf et al. "Audio-Based Multimedia Event Detection with DNNs and Sparse Sampling". In: *Proc. 5th ACM on International Conference on Multimedia Retrieval (ICMR)*. 2015, pp. 611–614.

[6] P. K. Atrey, N. C. Maddage, and M. S. Kankanhalli. "Audio Based Event Detection for Multimedia Surveillance". In: *Proc. 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings (ICASSP)*. 2006, pp. 813–816.

[7] J.-J. Aucouturier, B. Defreville, and F. Pachet. "The Bag-of-Frames Approach to Audio Pattern Recognition: A Sufficient Model for Urban Soundscapes But Not for Polyphonic Music". In: *The Journal of the Acoustical Society of America* 122 (2007), pp. 881–891.

[8] *AudioNet: Audio Annotation of Consumer-Produced Video*. http://MultimediaCommons.org/audionet.

[9] Y. Aytar, C. Vondrick, and A. Torralba. "SoundNet: Learning Sound Representations from Unlabeled Video". In: *Proc. 30th Annual Conference on Neural Information Processing Systems (NIPS)*. 2016, pp. 892–900.

[10] V. Barbosa et al. "Browsing Videos by Automatically Detected Audio Events". In: *Proc. 2011 IEEE International Conference on Computer as a Tool (EUROCON)*. 2011, pp. 1–4.

[11] D. Barchiesi et al. "Acoustic Scene Classification: Classifying Environments from the Sounds They Produce". In: *IEEE Signal Processing Magazine* 32.3 (2015), pp. 16–34.

[12] E. Benetos et al. “Detection of Overlapping Acoustic Events Using a Temporally Constrained Probabilistic Model”. In: *Proc. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016, pp. 6450–6454.

[13] E. Benetos, G. Lafay, and M. Lagrange. *DCASE2016 Task 2 Baseline*. Tech. rep. Detection, Classification of Acoustic Scenes, and Events 2016, 2016.

[14] S. Bengio, J. Weston, and D. Grangier. “Label Embedding Trees for Large Multi-Class Tasks”. In: *Proc. 24th Annual Conference on Neural Information Processing Systems (NIPS)*. 2010, pp. 163–171.

[15] S. Boll et al. “Development of a Multimodal Reminder System for Older Persons in Their Residential Home”. In: *Informatics for Health and Social Care* 35.3–4 (2010), pp. 104–124.

[16] A. L. Borker et al. “Vocal Activity as a Low Cost and Scalable Index of Seabird Colony Size”. In: *Conservation Biology* 28.4 (2014), pp. 1100–1108.

[17] A. S. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, 1994.

[18] L. Breiman. “Random Forest”. In: *Machine Learning* 45 (2001), pp. 5–32.

[19] D. Brezeale and D. J. Cook. “Automatic Video Classification: A Survey of the Literature”. In: *IEEE Trans. on Systems, Man, and Cybernetics* 38.3 (2008), pp. 416–430.

[20] F. Briggs, X. Z. Fern, and J. Irvine. “Multi-Label Classifier Chains for Bird Sound”. In: *arXiv:1304.5862* (2013).

[21] F. Briggs et al. “Acoustic Classification of Multiple Simultaneous Bird Species: A Multi-Instance Multi-Label Approach”. In: *Journal of the Acoustical Society of America* 131.6 (2012), pp. 4640–4650.

[22] T. Butko. “Feature Selection for Multimodal Acoustic Event Detection”. PhD thesis. Universitat Politecnica de Catalunya, 2011.

[23] T. Butko et al. “Improving Detection of Acoustic Events Using Audiovisual Data and Feature Level Fusion”. In: *Proc. 10th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2009, pp. 1147–1150.

[24] T. Butko et al. “Acoustic Event Detection Based on Feature-Level Fusion of Audio and Video Modalities”. In: *EURASIP Journal on Advances in Signal Processing* 2011, 485738 (2011).

[25] T. Butko et al. “Two-Source Acoustic Event Detection and Localization: Online Implementation in a Smart-Room”. In: *Proc. 19th European Signal Processing Conference (EUSIPCO)*. 2011, pp. 1317–1321.

[26] E. Cakir et al. “Polyphonic Sound Event Detection Using Multi Label Deep Neural Networks”. In: *Proc. 2015 International Joint Conference on Neural Networks (IJCNN)*. 2015, pp. 1–7.

[27] C. Canton-Ferrer et al. “Audiovisual Event Detection Towards Scene Understanding”. In: *Proc. 2009 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*. 2009, pp. 81–88.

[28] V. Carletti et al. “Audio Surveillance Using a Bag of Aural Words Classifier”. In: *Proc. 10th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*. 2013, pp. 81–86.

[29] R. Chakraborty and C. Nadeu. “Joint Recognition and Direction-of-Arrival Estimation of Simultaneous Meeting Room Acoustic Events”. In: *Proc. 14th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2013, pp. 2948–2952.

[30] M. Chan et al. “A Review of Smart Homes - Present State and Future Challenges”. In: *Computer Methods and Programs in Biomedicine* 91.1 (2008), pp. 55–81.

[31] C. Chang and C. Lin. “LIBSVM: A Library for Support Vector Machines”. In: *ACM Trans. on Intelligent Systems and Technology* 2.3 (2011). Article No. 27.

[32] S. Chaudhuri and B. Raj. “Unsupervised Structure Discovery for Semantic Analysis of Audio”. In: *Proc. 26th Annual Conference on Neural Information Processing Systems (NIPS)*. 2012, pp. 1178–1186.

[33] J. Chen et al. “An Automatic Ccoustic Bathroom Monitoring System”. In: *Proc. 2005 International Symposium on Circuits and Systems (ISCAS)*. 2005, pp. 1750–1753.

[34] H. Cheng et al. *SRI-Sarnoff AURORA system at TRECVID 2012: Multimedia event detection and recounting*. Tech. rep. TREC Video Retrieval Evaluation, 2012.

[35] M. L. Chin and J. J. Burred. “Audio Event Detection Based on Layered Symbolic Sequence Representations”. In: *Proc. 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2012, pp. 1520–6149.

[36] N. Cho and E Kim. “Enhanced Voice Activity Detection Using Acoustic Event Detection and Classification”. In: *IEEE Trans. on Consumer Electronics* 57.1 (2011), pp. 196–202.

[37] I. Choi et al. *DNN-Based Sound Event Detection with Exemplar-Based Approach for Noise Reduction*. Tech. rep. Detection, Classification of Acoustic Scenes, and Events 2016, 2016.

[38] K. Choi, G. Fazekas, and M. Sandler. “Explaining Deep Convolutional Neural Networks on Music Classification”. In: *arXiv:1607.02444* (2016).

[39] S. Chu, S. Narayanan, and C.-C. Kuo. “Environmental Sound Recognition With Time-Frequency Audio Features”. In: *IEEE Trans. on Audio, Speech, and Language Processing* 17.6 (2009), pp. 1142–1158.

[40] C. Clavel, T. Ehrette, and G. Richard. “Events Detection for an Audio-Based Surveillance System”. In: *Proc. 2005 IEEE International Conference on Multimedia and Expo (ICME)*. 2005, pp. 1306–1309.

[41] C. V. Cotton and D. P. W. Ellis. “Spectral vs. Spectro-Temporal Features for Acoustic Event Detection”. In: *Proc. 2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2011, pp. 69–72.

[42] A. Criminisi, J. Shotton, and E. Konukoglu. “Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning”. In: *Foundations and Trends in Computer Graphics and Computer Vision* 7.2–3 (2012), pp. 81–227.

[43] A. Criminisi et al. “Regression Forests for Efficient Anatomy Detection and Localization in Computed Tomography Scans”. In: *Medical Image Analysis* 17.8 (2013), pp. 1293–1303.

[44] E. Dafna, A. Tarasiuk, and Y. Zigel. “Automatic Detection of Whole Night Snoring Events Using Non-Contact Microphone”. In: *PLoS One* 8.12 (2013), e84139.

[45] J. Deng et al. “ImageNet: A Large-Scale Hierarchical Image Database”. In: *Proc. 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 248–255.

[46] L. Deng and X. Li. “Machine Learning Paradigms for Speech Recognition: An Overview”. In: *IEEE Trans. on Audio, Speech, and Language Processing* 21.5 (2013), pp. 1060–1089.

[47] J. Dennis, H. D. Tran, and H. Li. “Spectrogram Image Feature for Sound Event Classification in Mismatched Conditions”. In: *Signal Processing Letters* 18.2 (2011), pp. 130–13.

[48] J. Dennis. “Sound Event Recognition in Unstructured Environments using Spectrogram Image Processing”. PhD thesis. Nanyang Technological University, 2014.

[49] J. Dennis, H. D. Tran, and E. S. Chng. “Image Feature Representation of the Subband Power Distribution for Robust Sound Event Classification”. In: *IEEE Trans. on Audio, Speech, and Language Processing* 21.2 (2013), pp. 367–377.

[50] J. Dennis, H. D. Tran, and E. S. Chng. “Overlapping Sound Event Recognition Using Local Spectrogram Features and the Generalised Hough Transform”. In: *Pattern Recognition Letters* 34.9 (2013), pp. 1085–1093.

[51] J. Dennis et al. “Temporal Coding of Local Spectrogram Features for Robust Sound Recognition”. In: *Proc. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2013, pp. 803–807.

[52] A. Dessein, A. Cont, and G. Lemaitre. “Matrix Information Geometry”. In: Springer, 2013. Chap. Real-Time Detection of Overlapping Sound Events with Non-Negative Matrix Factorization, pp. 341–371.

[53] J. Dibiase, H. Silverman, and M. Brandstein. *Microphone Arrays. Robust Localization in Reverberant Rooms*. Springer, New York, 2001.

[54] O. Dikmen and A. Mesaros. “Sound Event Detection Using Non-Negative Dictionaries Learned from Annotated Overlapping Events”. In: *Proc. 2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2013, pp. 1–4.

[55] A. Diment, T. Heittola, and T. Virtanen. *Sound Event Detection for Office Live and Office Synthetic AASP Challenge*. Tech. rep. IEEE AASP Challenge on Detection, Classification of Acoustic Scenes, and Events, 2013.

[56] W. D. Duckitt, S. K. Tuomi, and T. R. Niesler. “Automatic Detection, Segmentation and Assessment of Snoring from Ambient Acoustic Data”. In: *Physiological Measurement* 27.10 (2006), pp. 1047–1056.

[57] S. R. Eddy. “Profile Hidden Markov Models”. In: *Bioinformatics Review* 14.9 (1998), pp. 755–763.

[58] B. Elizalde, M. Ravanelli, and G. Friedland. “Audio Concept Ranking for Video Event Detection on User-Generated Content”. In: *Proc. 1st Workshop on Speech, Language and Audio in Multimedia (SLAM)*. 2013, pp. 9–14.

[59] B. Elizalde et al. “Audio-Concept Features and Hidden Markov Models for Multimedia Event Detection”. In: *Proc. 2nd International Workshop on Speech, Language and Audio in Multimedia (SLAM)*. 2014, pp. 3–8.

[60] B. Elizalde et al. *Experimentation on The DCASE Challenge 2016: Task 1 - Acoustic Scene Classification and Task 3 - Sound Event Detection in Real Life Audio*. Tech. rep. Detection, Classification of Acoustic Scenes, and Events 2016, 2016.

[61] M. Espi et al. “Exploiting Spectro-Temporal Locality in Deep Learning Based Acoustic Event Detection”. In: *EURASIP Journal on Audio, Speech, and Music Processing* 2015.26 (2015).

[62] G. Evangelopoulos et al. “Video Event Detection and Summarization Using Audio, Visual and Text Saliency”. In: *Proc. 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2009, pp. 3553–3556.

[63] M. Ferroudj et al. “Detection of Rain in Acoustic Recordings of the Environment”. In: *Proc. 13th Pacific Rim International Conference on Artificial Intelligence (PRICAI)*. 2014, pp. 104–116.

[64] W. Fisher, G. Doddington, and K. Goudie-Marshall. “The DARPA Speech Recognition Research Database: Specifications and Status”. In: *Proc. DARPA Workshop on Speech Recognition*. 1986, pp. 93–99.

[65] P. Foggia et al. “Audio Surveillance of Roads: A System for Detecting Anomalous Sounds”. In: *IEEE Trans. on Intelligent Transportation Systems* 17.1 (2016), pp. 279–288.

[66] P. Foggia et al. “Reliable Detection of Audio Events in Highly Noisy Environments”. In: *Pattern Recognition Letters* 65 (2015), pp. 22–28.

[67] G. D. Forney. “Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference”. In: *IEEE Trans. Information Theory* . 18.3 (1972), pp. 363–378.

[68] M. Gales and S. Young. *The Application of Hidden Markov Models in Speech Recognition*. Vol. 1. Now Publishers Inc., Hanover, USA, 2008.

[69] J. Gall et al. “Hough Forests for Object Detection, Tracking, and Action Recognition”. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 33.11 (2011), pp. 2188–2202.

[70] V. Garg et al. “Privacy Concerns in Assisted Living Technologies”. In: *Annals of Telecommunications* 69 (2014), pp. 75–88.

[71] J. F. Gemmeke et al. “An Exemplar-Based NMF Approach for Audio Event Detection”. In: *Proc. Proc. 2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2013, pp. 1–4.

[72] P. Geurts, D. Ernst, and L. Wehenkel. “Extremely Randomized Trees”. In: *Machine Learning* 63.1 (2006), pp. 3–42.

[73] B. Ghoraani and S. Krishnan. “Time-Frequency Matrix Feature Extraction and Classification of Environmental Audio Signals”. In: *IEEE Tran. on Audio, Speech, and Language Processing* 19.7 (2011), pp. 2197–2209.

[74] D. Giannoulis et al. “A Database and Chanllenge for Acoustic Scene Classification and Event Detection”. In: *Proc. 21st European Signal Processing Conference (EUSIPCO)*. 2013, pp. 1–5.

[75] D. Giannoulis et al. “Detection and Classification of Acoustic Scenes and Events: An IEEE AASP Challenge”. In: *Proc. 2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2013, pp. 1–4.

[76] P. Giannoulis et al. “Multi-Microphone Fusion for Detection of Speech and Acoustic Events in Smart Spaces”. In: *Proc. 22nd European Signal Processing Conference (EUSIPCO)*. 2014, pp. 2375–2379.

[77] P. Giannoulis et al. *Improved Dictionary Selection and Detection Schemes in Sparse-CNMF-Based Overlapping Acoustic Event Detection*. Tech. rep. Detection, Classification of Acoustic Scenes, and Events 2016, 2016.

[78] M. Gnen and E. Alpaydin. “Multiple Kernel Learning Algorithms”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2211–2268.

[79] J. J. Godfrey, E. C. Holliman, and J. McDaniel. “SWITCHBOARD: Telephone Speech Corpus for Research and Development”. In: *Proc. 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vol. 1. 1992, pp. 517–520.

[80] S. Goetze et al. “Hands-Free Telecommunication for Elderly Persons Suffering from Hearing Deficiencies”. In: *Proc. 12th IEEE International Conference on E-Health Networking, Application and Services*. 2010, pp. 209–224.

[81] S. Goetze et al. “Acoustic Monitoring and Localization for Social Care”. In: *Journal of Computing Science and Engineering* 6.1 (2011), pp. 40–50.

[82] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[83] R. Gordon, ed. *Ethnologue: Languages of the World*. Dallas: SIL International, 2005.

[84] A. Gorin, N. Makhazhanov, and N. Shmyrev. *DCASE 2016 Sound Event Detection System Based on Convolutional Neural Network*. Tech. rep. Detection, Classification of Acoustic Scenes, and Events 2016, 2016.

[85] R. Grzeszick, A. Plinge, and G. A. Fink. “Temporal Acoustic Words for Online Acoustic Event Detection”. In: *Proc. 37th German Conference Pattern Recognition (GCPR)*. 2015, pp. 142–153.

[86] S. Gunasekaran and K. Revathy. “Content-Based Classification and Retrieval of Wild Animal Sounds Using Feature Selection Algorithm”. In: *Proc. 2nd International Conference on Machine Learning and Computing (ICMLC)*. 2010, pp. 272–275.

[87] H. Guo et al. “Tefnut: An Accurate Smartphone Based Rain Detection System in Vehicles”. In: *Proc. International Conference on Wireless Algorithms, Systems, and Applications (WASA)*. 2016, pp. 13–23.

[88] J. M. Gutierrez-Arriola et al. *Synthetic Sound Event Detection based on MFCC*. Tech. rep. Detection, Classification of Acoustic Scenes, and Events 2016, 2016.

[89] A. Hannun et al. “Deep Speech: Scaling up End-to-End Speech Recognition”. In: *arXiv:1412.5567* (2014).

[90] T. Hayashi et al. *Bidirectional LSTM-HMM Hybrid System for Polyphonic Sound Event Detection*. Tech. rep. Detection, Classification of Acoustic Scenes, and Events 2016, 2016.

[91] M. Heckmann. “Steps Towards More Natural Human-Machine Interaction via Audio-Visual Word Prominence Detection”. In: *Proc. International Workshop on Multimodal Analyses Enabling Artificial Agents in Human-Machine Interaction (MA3HMI)*. 2014, pp. 15–24.

[92] T. Heittola et al. “Audio Context Recognition Using Audio Event Histogram”. In: *Proc. 18th European Signal Processing Conference (EUSIPCO)*. 2010, pp. 1272–1276.

[93] T. Heittola et al. “Context-Dependent Sound Event Detection”. In: *EURASIP Journal on Audio, Speech, and Music Processing* 2013.1 (2013).

[94] T. Heittola et al. “Supervised Model Training for Overlapping Sound Events based on Unsupervised Source Separation”. In: *Proc. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2013, pp. 8677–8681.

[95] H. Hermansky, D. P. W. Ellis, and S. Sharma. “Tandem Connectionist Feature Extraction for Conventional HMM Systems”. In: *Proc. 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (ICASSP)*. Vol. 3. 2000, pp. 1635–1638.

[96] S. Hershey et al. “CNN Architectures for Large-Scale Audio Classification”. In: *arXiv:1609.09430* (2016).

[97] L. Hertel, H. Phan, and A. Mertins. “Comparing Time and Frequency Domain for Audio Event Recognition Using Deep Learning”. In: *Proc. 2016 International Joint Conference on Neural Networks (IJCNN)*. 2016, pp. 3407–3411.

[98] G. Hinton et al. “Deep Neural Networks for Acoustic Modeling in Speech Recognition”. In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 82–97.

[99] M. Hoai and F. De la Torre. “Max-Margin Early Event Detectors”. In: *Proc. 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012, pp. 2863–2870.

[100] C. Huang, C. C. Loy, and X. Tang. “Unsupervised Learning of Discriminative Attributes and Visual Representations”. In: *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 5175–5184.

[101] C.-J. Huang et al. “Frog Classification Using Machine Learning Techniques”. In: *Expert Systems with Applications* 36 (2009), pp. 3737–3743.

[102] D. Huang et al. “Sequential Max-Margin Event Detectors”. In: *Proc. European Conference on Computer Vision (ECCV)*. 2014, pp. 410–424.

[103] P.-S. Huang. “Non-Speech Acoustic Event Detection Using Multimodal Information”. PhD thesis. University of Illinois at Urbana-Champaign, 2012.

[104] P.-S. Huang, X. Zhuang, and M. Hasegawa-Johnson. “Improving Acoustic Event Detection using Generalizable Visual Features and Multi-Modality Modeling”. In: *Proc. 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2011, pp. 349–352.

[105] E. Humphrey, J. Bello, and Y. Lecun. “Moving Beyond Feature Design: Deep Architectures and Automatic Feature Learning in Music Informatics”. In: *Proc. 13th International Society of Music Information Retrieval Conference (ISMIR)*. 2012, pp. 403–408.

[106] E. J. Humphrey, J. P. Bello, and Y. LeCun. “Feature Learning and Deep Architectures: New Directions for Music Informatics”. In: *Journal of Intelligent Information Systems* 41.3 (2013), pp. 461–481.

[107] S. Ikbal and T. Faruquie. “HMM Based Event Detection in Audio Conversation”. In: *Proc. 2008 IEEE International Conference on Multimedia and Expo (ICME)*. 2008, pp. 1497–1500.

[108] K. Imoto and N. Ono. “Spatial-Feature-Based Acoustic Scene Analysis Using Distributed Microphone Array”. In: *Proc. 23rd European Signal Processing Conference (EUSIPCO)*. 2015, pp. 739–743.

[109] W. Jiang, C. Cotton, and A. C. Loui. “Automatic Consumer Video Summarization by Audio and Visual Analysis”. In: *Proc. 2011 IEEE International Conference on Multimedia and Expo (ICME)*. 2011, pp. 1–6.

[110] Y.-G. Jiang et al. *Columbia-UCF TRECVID2010 Multimedia Event Detection: Combining Multiple Modalities, Contextual Concepts, and Temporal Matching*. Tech. rep. TREC Video Retrieval Evaluation, 2010.

[111] Q. Jin et al. “Event-Based Video Retrieval Using Audio”. In: *Proc. 13th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2012, pp. 2085–2088.

[112] T. Kanungo et al. “An Efficient k-Means Clustering Algorithm: Analysis and Implementation”. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)* 24.7 (2002), pp. 881–892.

[113] T. L.M. V. Kasteren, G. Englebienne, and B. J. A. Kröse. “Hierarchical Activity Recognition Using Automatically Clustered Actions”. In: *Proc. 2nd International Conference on Ambient Intelligence (AmI)*. 2011, pp. 82–91.

[114] T. Komatsu et al. *Acoustic Event Detection Method Using Semi-Supervised Non-Negative Matrix Factorization with a Mixture of Local Dictionaries*. Tech. rep. Detection, Classification of Acoustic Scenes, and Events 2016, 2016.

[115] Q. Kong et al. *Deep Neural Network Baseline for DCASE Challenge 2016*. Tech. rep. Detection, Classification of Acoustic Scenes, and Events 2016, 2016.

[116] A. Kumar et al. “Audio Event Detection from Acoustic Unit Occurrence Patterns”. In: *Proc. 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2012, pp. 489–492.

[117] A. Kumar et al. “Event Detection in Short Duration Audio Using Gaussian Mixture Model and Random Forest Classifier”. In: *Proc. 21st European Signal Processing Conference (EUSIPCO)*. 2013, pp. 1–5.

[118] A. Kumar and B. Raj. “Audio Event Detection using Weakly Labeled Data”. In: *Proc. 2016 ACM on Multimedia Conference (ACMMM)*. 2016, pp. 1038–1047.

[119] A. Kumar and B. Raj. “Features and Kernels for Audio Event Recognition”. In: *arXiv:1607.05765* (2016).

[120] J. Kürby et al. “Bag-of-Features Acoustic Event Detection for Sensor Networks”. In: *Proc. Detection and Classification of Acoustic Scenes and Events (DCASE) Workshop*. 2016, pp. 55–59.

[121] P. Laffitte et al. “Deep Neural Networks for Automatic Detection of Screams and Shouted Speech in Subway Trains”. In: *Proc. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016, pp. 6460–6464.

[122] Y.-H. Lai et al. *DCASE Report for Task 3: Sound Event Detection in Real Life Audio*. Tech. rep. Detection, Classification of Acoustic Scenes, and Events 2016, 2016.

[123] S. Lazebnik, C. Schmid, and J. Ponce. “Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories”. In: *Proc. 2006 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2. 2006, pp. 2169–2178.

[124] Y. LeCun, Y. Bengio, and G. Hinton. “Deep Learning”. In: *Nature* 521.7553 (2015), pp. 436–444.

[125] H. Lee et al. “Unsupervised Feature Learning for Audio Classification Using Convolutional Deep Belief Networks”. In: *Proc. 23rd Annual Conference on Neural Information Processing Systems (NIPS)*. 2009, pp. 1096–1104.

[126] J. Lee et al. “Stress Detection and Classification of Laying Hens by Sound Analysis”. In: *Asian-Australasian Journal of Animal Sciences* 28.4 (2015), pp. 592–598.

[127] K. F. Lee and H. W. Hon. “Speaker-Independent Phone Recognition Using Hidden Markov Models”. In: *IEEE Trans. on Acoustics, Speech and Signal Processing* 37.11 (1989), pp. 1641–1648.

[128] K. Lee and D. P. W. Ellis. “Audio-Based Semantic Concept Classification for Consumer Video”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.6 (2010), pp. 1406–1416.

[129] Y. Lee, D. K. Han, and H. Ko. "Acoustic Signal Based Abnormal Event Detection in Indoor Environment Using Multiclass Adaboost". In: *Proc. 2013 IEEE International Conference on Consumer Electronics (ICCE)*. 2013, pp. 322–323.

[130] B. Leibe, A. Leonardis, and B. Schiele. "Robust Object Detection with Interleaved Categorization and Segmentation". In: *International Journal of Computer Vision* 77.1–3 (2008), pp. 259–289.

[131] A. Lerch. "In An Introduction to Audio Content Analysis". In: Wiley-IEEE Press, 2012. Chap. Instantaneous Features, pp. 31–69.

[132] J. Liu et al. *SRI-Sarnoff AURORA system at TRECVID 2013: Multimedia event detection and recounting.* Tech. rep. TREC Video Retrieval Evaluation, 2013.

[133] X. Lu et al. "Sparse Representation Based on a Bag of Spectral Exemplars for Acoustic Event Detection". In: *Proc. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014, pp. 6255–6259.

[134] X. Lu et al. "Spectral Patch Based Sparse Coding for Acoustic Event Detection". In: *Proc. 9th International Symposium on Chinese Spoken Language Processing (ISCSLP 2014)*. 2014, pp. 317–320.

[135] R. F. Lyon. "Machine Hearing: An Emerging Field". In: *IEEE Signal Processing Magazine* 27.5 (2010), pp. 131–139.

[136] R. F. Lyon et al. "Sound Retrieval and Ranking Using Sparse Auditory Representations". In: *Neural Computation* 22.9 (2010), pp. 2390–2416.

[137] S. Maji, A. C. Berg, and J. Malik. "Efficient Classification for Additive Kernel SVMs". In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 35.1 (2013), pp. 66–77.

[138] T. A. Marques et al. "Estimating Animal Population Density Using Passive Acoustics". In: *Biological Reviews* 88.2 (2013), pp. 287–309.

[139] J. Maxime et al. "Sound Representation and Classification Benchmark for Domestic Robots". In: *Proc. 2014 IEEE International Conference on Robotics and Automation (ICRA)*. 2014, pp. 6285–6292.

[140] I. McLoughlin et al. "Robust Sound Event Classification using Deep Neural Networks". In: *IEEE/ACM Trans. on Audio, Speech, and Language Processing* 23.3 (2015), pp. 540–552.

[141] M. Meis et al. "Telemonitoring and Assistant System for People with Hearing Deficiencies: First Results from a User Requirement Study". In: *Proc. European Conference on eHealth (ECEH)*. 2007, pp. 163–175.

[142] M. Memon et al. "Ambient Assisted Living Healthcare Frameworks, Platforms, Standards, and Quality Attributes". In: *Sensors* 14.3 (2014), pp. 4312–4341.

[143] A. Mesaros, T. Heittola, and T. Virtanen. "TUT Database for Acoustic Scene Classification and Sound Event Detection". In: *Proc. 24th European Signal Processing Conference (EUSIPCO)*. 2016, pp. 1128–1132.

[144] A. Mesaros et al. "Acoustic Event Detection in Real Life Recordings". In: *Proc. 18th European Signal Processing Conference (EUSIPCO)*. 2010, pp. 1267–1271.

[145] A. Mesaros et al. "Sound Event Detection in Real Life Recordings Using Coupled Matrix Factorization of Spectral Representations and Class Activity Annotations". In: *Proc. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 151–155.

[146] D. Mitrovic and M. Zeppelzauer. "Discrimination and Retrieval of Animal Sounds". In: *Proc. 12th International Multi-Media Modelling Conference (MMM)*. 2006, p. 5.

[147] F. Moosmann, B. Triggs, and F. Jurie. "Fast Discriminative Visual Codebooks using Randomized Clustering Forests". In: *Proc. 20th Annual Conference on Neural Information Processing Systems (NIPS)*. 2006, pp. 985–992.

[148] C. Müller et al. "Speech-Overlapped Acoustic Event Detection for Automotive Applications". In: *Proc. 9th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2008, pp. 2590–2593.

[149] F. Müller and A. Mertins. "Contextual Invariant-Integration Features for Improved Speaker-Independent Speech Recognition". In: *Speech Communication* 53.6 (2011), pp. 830–841.

[150] P. Muthukumar and A. W. Black. "A Deep Learning Approach to Data-driven Parameterizations for Statistical Parametric Speech Synthesis". In: *arXiv:1409.8558* (2014).

[151] C. Nadeu, R. Chakraborty, and M. Wolf. "Model-Based Processing for Acoustic Scene Analysis". In: *Proc. 22nd European Signal Processing Conference (EUSIPCO)*. 2014, pp. 2370–2374.

[152] C. Nadeu, D. Macho, and J. Hernando. "Frequency and Time Filtering of Filter-Bank Energies for Robust HMM Speech Recognition". In: *Speech Communication* 34 (2001), pp. 93–114.

[153] S. Nakamura et al. "Acoustical Sound Database in Real Environments for Sound Scene Understanding and Handsfree Speech Recognition". In: *Proc. 2nd International Conference on Language Resources and Evaluation (ICLRE)*. Vol. 2. 2000, pp. 965–968.

[154] H. Ney and S. Ortmanns. "Dynamic Programming Search for Continuous Speech Recognition". In: *IEEE Signal Processing Magazine* 16 (1999), pp. 64–83.

[155] A. Y. Ng, M. I. Jordan, and Y. Weiss. "On Spectral Clustering: Analysis and an Algorithm". In: *Proc. Advances in Neural Information Processing Systems 14 (NIPS)*. 2001, pp. 849–856.

[156] V. Q. Nguyen et al. "Real-Time Audio Surveillance System for PTZ Camera". In: *Proc. International Conference on Advanced Technologies for Communications (ATC)*. 2013, pp. 392–397.

[157] M. E. Niessen, T. L.M. V. Kasteren, and A. Merentitis. *Hierarchical Sound Event Detection*. Tech. rep. IEEE AASP Challenge on Detection, Classification of Acoustic Scenes, and Events, 2013.

[158] W. Nogueira, G. Roma, and P. Herrera. *Automatic Event Classification Using Front End Single Channel Noise Reduction, MFCC Features and a Support Vector Machine Classifier*. Tech. rep. IEEE AASP Challenge on Detection, Classification of Acoustic Scenes, and Events, 2013.

[159] V. Panayotov et al. "LIBRISPEECH: An ASR Corpus based on Public Domain Audio Books". In: *Proc. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 5206–5210.

[160] S. Pancoast and M. Akbacak. "Bag-of-Audio-Words Approach for Multimedia Event Classification". In: *Proc. 14th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2013, pp. 2105–2108.

[161] S. Pancoast and M. Akbacak. "N-Gram Extension for Bag-of-Audio-Words". In: *Proc. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2013, pp. 778–782.

[162] S. Pancoast and M. Akbacak. "Softening Quantization in Bag-of-Audio-Words". In: *Proc. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014, pp. 1370–1374.

[163] G. Parascandolo, H. Huttunen, and T. Virtanen. "Recurrent Neural Networks for Polyphonic Sound Event Detection in Real Life Recordings". In: *Proc. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016, pp. 6440–6444.

[164] D. B. Paul and J. M. Baker. "The Design for the Wall Street Journal-Based CSR Corpus". In: *Proc. Workshop on Speech and Natural Language (HLT)*. 1991, pp. 357–362.

[165] G. Peeters. *A Large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO Project*. Tech. rep. 2004.

[166] S. Petridis, M. Leveque, and M. Pantic. "Audiovisual Detection of Laughter in Human-Machine Interaction". In: *Proc. 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII)*. 2013, pp. 129–134.

[167] H. Phan and A. Mertins. “Exploring Superframe Co-occurrence for Acoustic Event Recognition”. In: *Proc. 22nd European Signal Processing Conference (EUSIPCO)*. 2014, pp. 631–635.

[168] H. Phan et al. “A Multi-Channel Fusion Framework for Audio Event Detection”. In: *Proc. 2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2015, pp. 1–5.

[169] H. Phan et al. “Audio Phrases for Audio Event Recognition”. In: *Proc. 23rd European Signal Processing Conference (EUSIPCO)*. 2015, pp. 2546–2550.

[170] H. Phan et al. “Early Event Detection in Audio Streams”. In: *Proc. 2015 IEEE International Conference on Multimedia and Expo (ICME 2015)*. 2015, pp. 1–6.

[171] H. Phan et al. “Learning Compact Structural Representations for Audio Events Using Regressor Banks”. In: *Proc. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016, pp. 211–215.

[172] H. Phan et al. “Learning Representations for Nonspeech Audio Events through Their Similarities to Speech Patterns”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24.4 (2016), pp. 807–822.

[173] H. Phan et al. “Random Regression Forests for Acoustic Event Detection and Classification”. In: *IEEE/ACM Trans. on Audio, Speech, and Language Processing* 23.1 (2015), pp. 20–31.

[174] H. Phan et al. “Representing Nonspeech Audio Signals through Speech Classification Models”. In: *Proc. 16th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2015, pp. 3441–3445.

[175] H. Phan et al. “Acoustic Event Detection and Localization with Regression Forests”. In: *Proc. 15th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2014, pp. 2524–2528.

[176] H. Phan et al. “CaR-FOREST: Joint Classification-Regression Decision Forests for Overlapping Audio Event Detection”. In: *arXiv:1607.02306* (2016).

[177] K. J. Piczak. “ESC: Dataset for Environmental Sound Classification”. In: *Proc. 23rd ACM International Conference on Multimedia (ACMMM)*. 2015, pp. 1015–1018.

[178] K. J. Piczak. “Envoronmental Sound Classification with Convolutional Neural Networks”. In: *Proc. 2015 IEEE Internationl Workshop on Machine Learning for Signal Processing (MLSP)*. 2015, pp. 1–6.

[179] A. Pikrakis and Y. Kopsinis. *Dictionary Learning Assisted Template Matching for Audio Event Detection (Legato)*. Tech. rep. Detection, Classification of Acoustic Scenes, and Events 2016, 2016.

[180] A. Plinge, R. Grzeszick, and G. Fink. "A Bag-of-Features Approach to Acoustic Event Detection". In: *Proc. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014, pp. 3704–3708.

[181] J. Portelo et al. "Non-Speech Audio Event Detection". In: *Proc. 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2009, pp. 1973–1976.

[182] L. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[183] R. Radhakrishnan, A. Divakaran, and P. Smaragdis. "Audio Analysis for Surveillance Applications". In: *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2005, pp. 158–161.

[184] M. Ravanelli et al. "Audio Concept Classification with Hierarchical Deep Neural Networks". In: *Proc. 22nd European Signal Processing Conference (EUSIPCO)*. 2014, pp. 606–610.

[185] N. Razavi, J. Gall, and L. Van Gool. "Scalable Multi-Class Object Detection". In: *Proc. 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011, pp. 1505–1512.

[186] M. J. Reyes-Gomez and D. P. W. Ellis. "Selection, Parameter Estimation, and Discriminative Training of Hidden Markov Models for General Audio Modeling". In: *Proc. 2003 International Conference on Multimedia and Expo (ICME)*. Vol. 1. 2003, pp. I –73–6.

[187] D. A. Reynolds and R. C. Rose. "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models". In: *IEEE Trans. on Speech and Audio Processing* 3.1 (1995), pp. 72–83.

[188] G. Riegler et al. "Hough Networks for Head Pose Estimation and Facial Feature Localization". In: *Proc. British Machine Vision Conference (BMVC)*. 2014.

[189] M. A. Sadeghi and A. Farhadi. "Recognition Using Visual Phrases". In: *Proc. 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011, pp. 1745–1752.

[190] A. Saggese et al. "Time-Frequency Analysis for Audio Event Detection in Real Scenarios". In: *Proc. 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. 2016, pp. 438–443.

[191] T. N. Sainath et al. "Learning the Speech Front-End with Raw Waveform CLDNNs". In: *Proc. 16th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2015, pp. 1–5.

[192] J. Salamon, C. Jacoby, and J. P. Bello. "A Dataset and Taxonomy for Urban Sound Research". In: *Proc. 22nd ACM International Conference on Multimedia (ACMMM)*. 2014, pp. 1041–1044.

[193] M. Schmitt et al. “A Bag-of-Audio-Words Approach for Snore Sounds’ Excitation Localisation”. In: *Proc. ITG Symposium Speech Communication*. 2016, pp. 1–5.

[194] J. Schröder, J. Anemiiller, and S. Goetze. “Classification of Human Cough Signals Using Spectro-Temporal Gabor Filterbank Features”. In: *Proc. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016, pp. 6455–6459.

[195] J. Schröder, S. Goetze, and J. Anemüller. “Spectro-Temporal Gabor Filterbank Features for Acoustic Event Detection”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.12 (2015), pp. 2198–2208.

[196] J. Schröder et al. “Acoustic Event Detection Using Signal Enhancement and Spectro-Temporal Feature Extraction”. In: *Proc. 2013 IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (WASPAA)*. 2013, pp. 1–4.

[197] J. Schröder et al. “Ambient Assisted Living”. In: Springer, 2011. Chap. Detection and Classification of Acoustic Events for In-Home Care, pp. 181–195.

[198] S. Schulter et al. “Accurate Object Detection with Joint Classification-Regression Random Forests”. In: *Proc. 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014, pp. 923–930.

[199] S. Schulter et al. “Alternating Decision Forests”. In: *Proc. 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013, pp. 508–515.

[200] T. Schultz, N. T. Vu, and T. Schlippe. “GlobalPhone: A Multilingual Text & Speech Database in 20 Languages”. In: *Proc. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2013, pp. 8126–8130.

[201] M. A. Sehili et al. “Sound Environment Analysis in Smart Home”. In: *Proc. International Joint Conference on Ambient Intelligence (AmI)*. 2012, pp. 208–223.

[202] I. Sobieraj and M. D. Plumbley. *Coupled Sparse NMF vs. Random Forest Classification for Real Life Acoustic Event Detection*. Tech. rep. Detection, Classification of Acoustic Scenes, and Events 2016, 2016.

[203] N. Srivastava, E. Mansimov, and R. Salakhutdinov. “Unsupervised Learning of Video Representations using LSTMs”. In: *Proc. 32nd International Conference on Machine Learning (ICML)*. 2015, pp. 843–852.

[204] R. Stiefelhagen et al. “The CLEAR 2007 Evaluation”. In: *Multimodal Technologies for Perception of Humans*. 2009, pp. 3–34.

[205] J. A. Stork et al. “Audio-Based Human Activity Recognition Using Non-Markovian Ensemble Voting”. In: *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 2012, pp. 509–514.

[206] D. Stowell and M. D. Plumbley. “Automatic Large-Scale Classification of Bird Sounds is Strongly Improved by Unsupervised Feature Learning”. In: *PeerJ* 2 (2014), e488.

[207] D. Stowell et al. “Bird Detection in Audio: A Survey and a Challenge”. In: *CoRR abs/1608.03417* (2016).

[208] D. Stowell et al. “Detection and Classification of Acoustic Scenes and Events”. In: *IEEE Trans. on Multimedia* 17.10 (2015), pp. 1733–1746.

[209] C. Y. Suen. “N-Gram Statistics for Natural Language Understanding and Text Processing”. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 1.2 (1979), pp. 164–172.

[210] N. Takahashi, M. Gygli, and L. V. Gool. “AENet: Learning Deep Audio Features for Video Analysis”. In: *arXiv:1701.00599* (2017).

[211] N. Takahashi et al. “Deep Convolutional Neural Networks and Data Augmentation for Acoustic Event Recognition”. In: *Proc. 17th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2016, pp. 2982–2986.

[212] A. Temko and C. Nadeu. “Acoustic Event Detection in Meeting-Room Environments”. In: *Pattern Recognition Letters* 30 (2009), pp. 1281–1288.

[213] A. Temko, C. Nadeu, and J.-I. Biel. “Acoustic Event Detection: SVM-Based System and Evaluation Setup in CLEAR’07”. In: *Lecture Notes in Computer Science* 4625 (2008), pp. 354–363.

[214] A. Temko et al. “CLEAR Evaluation of Acoustic Event Detection and Classification Systems”. In: *Lecture Notes in Computer Science* 4122 (2007), pp. 311–322.

[215] A. Temko et al. “Computers in the Human Interaction Loop”. In: ed. by A. Waibel and R. Stiefelhagen. Springer London, 2009. Chap. Acoustic Event Detection and Classification, pp. 61–73.

[216] A. Temko. “Acoustic Event Detection and Classification”. PhD thesis. Universitat Politècnica de Catalunya, 2007.

[217] B. Thomee et al. “YFCC100M: The New Data in Multimedia Research”. In: *Communications of the ACM* 59 (2016), pp. 64–73.

[218] L. Torresani, M. Szummer, and A. Fitzgibbon. “Learning Query-Dependent Prefilters for Scalable Image Retrieval”. In: *Proc. 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 2615–2622.

[219] D. Tran et al. “Learning Spatiotemporal Features with 3D Convolutional Networks”. In: *Proc. IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 4489–4497.

[220] H. D. Tran and H. Li. “Jump Function Kolmogorov for Overlapping Audio Event Classification”. In: *Proc. 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2011, pp. 3696–3699.

[221] H. D. Tran and H. Li. “Sound Event Recognition With Probabilistic Distance SVMs”. In: *IEEE Trans. Audio, Speech, and Language Processing* 19.6 (2011), pp. 1556–1568.

[222] I. Trancoso et al. “Training Audio Events Detectors with a Sound Effects Corpus”. In: *Proc. 9th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2008.

[223] D. Ubskii and A. Pugachev. *Sound Event Detection in Real-Life Audio*. Tech. rep. Detection, Classification of Acoustic Scenes, and Events 2016, 2016.

[224] M. Vacher et al. “Challenges in the Processing of Audio Channels for Ambient Assisted Living”. In: *Proc. 12th IEEE International Conference on e-Health Networking Applications and Services (Healthcom)*. 2010, pp. 330–337.

[225] M. Vacher et al. “Development of Audio Sensing Technology for Ambient Assisted Living: Applications and Challenges”. In: *International Journal of E-Health and Medical Communications* 2.1 (2011), pp. 35–54.

[226] M. Valera and S. A. Velastin. “Intelligent Distributed Surveillance Systems: A Review”. In: *IEE Proceedings - Vision, Image and Signal Processing* 152.2 (2005), pp. 192–204.

[227] X. Valero and F. Alías. “Gammatone Cepstral Coefficients: Biologically Inspired Features for Non-Speech Audio Classification”. In: *IEEE Trans. Multimedia* 17.6 (2012), pp. 1684–1689.

[228] X. Valero and F. Alias. “Gammatone Wavelet Features for Sound Classification in Surveillance Applications”. In: *Proc. 20th European Signal Processing Conference (EUSIPCO)*. 2012, pp. 1658–1662.

[229] T. H. Vu and J.-C. Wang. *Acoustic Scene and Event Recognition Using Recurrent Neural Networks*. Tech. rep. Detection, Classification of Acoustic Scenes, and Events 2016, 2016.

[230] L. Vuegen et al. *An MFCC-GMM Approach for Event Detection and Classification*. Tech. rep. IEEE AASP Challenge on Detection, Classification of Acoustic Scenes, and Events, 2013.

[231] D. Wang and G. J. Brown. *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Wiley-IEEE Press, 2006.

[232] X. Wang and A. Gupta. “Unsupervised Learning of Visual Representations Using Videos”. In: *Proc. 2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 2794–2802.

[233] Y. Wang, S. Rawat, and F. Metze. “Exploring Audio Semantic Concepts for Event-Based Video Retrieval”. In: *Proc. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014, pp. 1360–1364.

[234] Y. Wang, L. Neves, and F. Metze. “Audio-Based Multimedia Event Detection Using Deep Recurrent Neural Networks”. In: *Proc. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016, pp. 2742–2746.

[235] L. R. Welch. “Hidden Markov Models and the Baum–Welch Algorithm”. In: *IEEE Information Theory Society Newsletter* 53.4 (2003).

[236] J. Xie et al. “Acoustic Classification of Australian Anurans Using Syllable Features”. In: *Proc. IEEE 10th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. 2015, pp. 1–6.

[237] J. Xie et al. “Detection of Anuran Calling Activity in Long Field Recordings for Bio-Acoustic Monitoring”. In: *Proc. IEEE 10th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. 2015, pp. 1–6.

[238] S. Young et al. *The HTK Book (for HTK Version 3.4.1)*. Tech. rep. Cambridge University Engineering Department, Cambridge, UK, 2009.

[239] W. Zajdel et al. “CASSANDRA: Audio-Video Sensor Fusion for Aggression Detection”. In: *Proc. 2007 IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS)*. 2007, pp. 200–205.

[240] H. Zhang, I. McLoughlin, and Y. Song. “Robust Sound Event Recognition Using Convolutional Neural Networks”. In: *Proc. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 559–563.

[241] J. Zhang et al. “Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study”. In: *International Journal of Computer Vision* 73.2 (2007), pp. 213–238.

[242] X. Zhou et al. “HMM-Based Acoustic Event Detection with AdaBoost Feature Selection”. In: *Lecture Notes in Computer Science* 4625 (2008), pp. 345–353.

[243] X. Zhuang et al. “Acoustic Fall Detection Using Gaussian Mixture Models and GMM Supervectors”. In: *Proc. 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2009, pp. 69–72.

[244] X. Zhuang et al. “Real-World Acoustic Event Detection”. In: *Pattern Recognition Letters* 31.12 (2010), pp. 1543–1551.

[245] C. Zieger. “An HMM Based System for Acoustic Event Detection”. In: *Lecture Notes in Computer Science* 4625 (2008).

[246] C. Zieger and M. Omologo. “Acoustic Event Classification Using a Distributed Microphone Network with a GMM/SVM Combined Algorithm”. In: *Proc. 9th Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2008, pp. 115–118.

[247] C. Zieger and M. Omologo. *Acoustic Event Detection - ITC-Irst AED Database*. Tech. rep. Internal ITC report, 2005.

[248] M. Zöhrer and F. Pernkopf. *Gated Recurrent Networks Applied To Acoustic Scene Classification and Acoustic Event Detection*. Tech. rep. Detection, Classification of Acoustic Scenes, and Events 2016, 2016.

[249] A. van den Oord, S. Dieleman, and B. Schrauwen. “Deep Content-Based Music Recommendation”. In: *Proc. 27th Annual Conference on Neural Information Processing Systems (NIPS)*. 2013, pp. 2643–2651.

[250] A. van den Oord et al. “WaveNet: A Generative Model for Raw Audio”. In: *arXiv:1609.03499* (2016).