OPTIMAL FEATURE SELECTION AND MACHINE LEARNING FOR HIGH-LEVEL AUDIO CLASSIFICATION

- A RANDOM FORESTS APPROACH

Muhammad Mazin Al-Maathidi

School of Computing, Science and Engineering
University of Salford, Salford, UK

Submitted in Partial Fulfilment of the Requirement of the Degree of Doctor of Philosophy, July 2017

Table of Contents

List of Figures

List of Tables

Acknowledgment

I would like to express my sincere gratitude to my supervisor Dr. Francis Li for his continuous support during the Ph.D study, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me throughout research and writing. Without his invaluable support and encouragement, this work would not have been accomplished. He has a great contribution to every progress I have made.

I especially thank my family members; my parents, my wife and my sisters. I would have never completed this without their support.

Finally, I would like to thank all people who prayed for me and supported me during my Ph.D study.

List of Abbreviations

Abbreviations	Meaning
AFF	Audio Fundamental Frequency
AH	Audio Harmonicity
AP	Audio Power
ASC	Audio Spectrum Centroid
ASE	Audio Spectrum Envelope
ASF	Audio Spectrum Flatness
ASP	Audio Spectrum Projection
ASS	Audio Spectrum Spread
BT	Bagged Tree
FFT	Fast Fourier Transform
GMM	Gaussian Mixture Models
HR	Harmonic Ratio
HSC	Harmonic Spectral Centroid
HSD	Harmonic Spectral Deviation
HSS	Harmonic Spectral Spread
HSV	Harmonic Spectral Variation
k-NN	*k*-Nearest Neighbours
LAT	Log Attack Time
LLD	Low Level Descriptors
MaxR	Maximum Range
MFCC	Mel-Frequency Cepstral Coefficients
MinR	Minimum Range
MPEG	Moving Picture Experts Group
NNet	Neural Network
PCA	Principal Component Analysis
RFs	Random Forests
SVD	Singular Value Decomposition
SVM	Support Vector Machines
TC	Temporal Centroid
ULH	Upper Limit of Harmonicity
ZCR	Zero-Crossing Rate

Abstract

Content related information, metadata, and semantics can be extracted from soundtracks of multimedia files. Speech recognition, music information retrieval and environmental sound detection techniques have been developed into a fairly mature technology enabling a final text mining process to obtain semantics for the audio scene. An efficient speech, music and environmental sound classification system, which correctly identify these three types of audio signals and feed them into dedicated recognisers, is a critical pre-processing stage for such a content analysis system. The performance and computational efficiency of such a system is predominately dependent on the selected features.

This thesis presents a detailed study to identify the suitable classification features and associate a suitable machine learning technique for the intended classification task. In particular, a systematic feature selection procedure is developed to employ the random forests classifier to rank the features according to their importance and reduces the dimensionality of the feature space accordingly. This new technique avoids the trial-and-error approach used by many authors researchers. The implemented feature selection produces results related to individual classification tasks instead of the commonly used statistical distance criteria based approaches that does not consider the intended classification task, which makes it more suitable for supervised learning with specific purposes. A final collective decision-making stage is employed to combine multiple class detectors patterns into one to produce a single classification result for each input frames.

The performance of the proposed feature selection technique has been compared with the techniques proposed by MPEG-7 standard to extract the reduced feature space. The results show a significant improvement in the resulted classification accuracy, at the same time, the feature space is simplified and computational overhead reduced.

The proposed feature selection and machine learning technique enable the use of only 30 out of the 47 features without degrading the classification accuracy while the classification accuracy lowered by 1.7% only while just 10 features were utilised. The validation shows good performance also and the last stage of collective decision making was able to improve the classification result even after selecting only a small number of classification features. The work represents a successful attempt to determine audio feature importance and classify the audio contents into speech, music and environmental sound using a selected feature subset. The result shows a high degree of accuracy by utilising the random forests for both feature importance ranking and audio content classification.

Chapter 1: Introduction

1.1 Introduction

Over the past century, since the invention of recording techniques, a large number of archives of recorded audio, video and movies has been accumulated. Digitisation can preserve or even enhance the content, but the usefulness of the archives relies heavily upon the availability of an effective indexing, search and data mining tools.

Digital media and user-contributed content, in recent years, has led to a further growth in the volume of media archives, which demands effective content analysis and data mining systems to manage the content and archives. Soundtracks are information rich and there is a considerable amount of related information that can be extracted to help to describe their content. Manual indexing and metadata tagging are time consuming and can be biased on some occasions due to individual workers.

Soundtracks can be categorised into three classes: speech, music, and environmental sound. These classes are closely related to the following existing audio information-mining techniques:

1. Automated speech recognition: This has developed into a mature technique, as evidenced by the availability of audio dictation tools.
2. Music information retrieval: This has achieved significant progress, allowing not only automated music transcription and analysis but also advanced analysis.
3. Environmental sound detection: This was developed mostly for environmental noise classification, but can be used for scene analysis.

To redeploy these tools for an efficient audio content analysis, it is essential to have a pre-processing stage that accurately classifies and timestamps where speech, music and environmental sounds occur, in order to apply the dedicated algorithms accordingly (Li, 2013).

Commonly used feature spaces include Mel-frequency cepstral coefficients (MFCC), sub-band energy ratio, zero-crossing rates (ZCRs) etc. The classifications moreover are typically based upon supervised learning models such as artificial neural networks (NNets), support vector machine (SVM), *k*-nearest neighbours (k-NN), Classification Trees and Random Forests (RFs). For detailed speech modelling, especially for speech recognition, probabilistic models such as hidden Markov models (HMMs) and Gaussian mixture model (GMM) are found suitable. For example, Liu et al. (2007) propose an audio-classification and speaker-identification method, in which the MFCC, sub-band energy ratio and zero-crossing rates are used as feature spaces; SVMs are adopted for general audio classification and GMMs for further speaker identification. Researchers develop individual systems for specific purposes and choose audio features in an ad hoc or empirical way. It seems that little has been done to systematically and effectively rank the audio features according to their contribution to the intended classification task and select the top ranked features as an optimal classification features for that particular classification problem, as evidenced by a lack of existing literature. This thesis addresses these issues and proposes the use of random forests to effectively prioritise features and decision trees to classify audio content into speech, music and environmental sound.

1.2 Aim of the Research

This research aims at developing a systematic approach for feature selection based on feature importance ranking, and identifying a more suitable machine learning approach for audio classification into three categories namely speech, music and environmental sound. The feature selection approach is based on RFs in order to test the hypothesis that RFs present a suitable machine learning tool for the aforementioned audio classification problem.

The proposed technique consists of a set of systematic methods that utilise the RFs to rank the features according to their importance for the intended classification tasks. Ranking will prepare the utilised feature list for the backward feature selection in order to select the smallest possible subset of highly related features. Wrapper method combined with classification accuracy measure is used to evaluate the efficiency of the selected features subset.

RFs feature ranking provides a direct indicator of the contribution made by each feature to attain the final classification result, feature selection will utilise this ranking to produce the smallest possible features that facilitate the optimum classification performance.

The proposed technique will reduce the computational load during the feature extraction phase by calculating only the critical features that have a direct contribution towards the final classification result. The required classification time will be reduced also by simplifying the mapping problem between the input features and the output classification result.

The classification of the aforementioned classes is performed in two phases. The first phase converts the multiple class classification task into multiple binary class detectors. The second phase involves a collective decision making that combines the result of the three binary classifiers to a single classification result per frame. The results of the multiple class detectors are processed by a collective decision-making stage to produce the final classification result that contains the predicted classes label of the input audio file frames.

1.3 Scope of the Study

This study considers three classes: speech, music, and environmental sound to test and evaluate the proposed feature selection technique. The samples have pure class content without overlapping between the classes. The case of pure and non-overlapped audio content was

adopted to enable the smooth development and to evaluate the feature selection technique. The other reason of the utilisation of such a data set is that the content of these three classes are intended to be further processed by more dedicated semantic classifiers -as discussed in Chapter 2- and the semantic classifiers need a good quality audio content to be able to accomplish a good semantic analysis. This would pave the way for the development of other audio content classification techniques that study more detailed audio classes.

1.4 The MPEG-7 Standard

MPEG-7 is an ISO/IEC multimedia content description standard that was developed by MPEG (Moving Picture Experts Group). It provides a rich set of standardisation tools to describe multimedia content including audio, images and video data. It utilises eXtensible Markup Language (XML) to store the metadata and utilise time coding by synchronising the description with specific intervals through the multimedia files (Salembier and Sikora, 2002; Kim et al., 2005).

The main elements in the MPEG-7 Standard are (Kim et al., 2005):

- Descriptors: they outline the syntax and the semantics definition of audio feature vectors and their elements.
- Description schemes: they specify the structure and semantics of the relationships between the components.
- A description definition language: it is used to define the syntax of existing or new MPEG-7 description tools.
- System tools: these tools deal with the binary-coded representation of descriptors or description schemes to enable an efficient storage, transmission and multiplexing.

The MPEG-7 content description includes many categories of information. The following list shows a sample of categories and lists few examples for each category (Kim et al., 2005).

- Production related information (title, author, composer).
- Content usage information (copyright, usage history, broadcast information).
- Storage features (file format, encoding).
- Low level features (spectral descriptors, harmony descriptors).
- Conceptual information (objects, events, objects interactions).
- User interaction information (user preferences, usage history).

After this review of the MPEG-7 standard, it becomes clear that the scope of the study is highly related to MPEG-7 standard. For this reason, the MPEG-7.

1.5 Research Outcomes

This is an attempt to apply random forests to determine the feature importance and decision trees to classify sound tracks into speech, music and environmental sound. Attempts of applying RFs to audio data are relatively new and have just recently been reported. Auditory contexts which are the acoustic scene of specific location or site such as a restaurant or bus station (Yang and Su, 2012), and environmental sound data analysis (Zhang and Lv, 2015). These indicate that the random forests were effective in handling environmental sound classification. It is also surprising to note that there was no reported effort of utilising the random forests features ranking to perform feature selection and audio classification into speech, music and environmental sound.

The features used for environmental sound and soundscapes analysis are generally different from those used for speech, music and event/or effect sound commonly seen in the sound tracks of media files. For the latter MPEG LLD are becoming more prevalent and

popular, due to the MPEG-7 standardisation. Different sources and diverse features used mean that the classification behaviour may substantively different. The results from the literature cannot be generalised to speech, music and environmental sound classification or discrimination. Therefore, it will be interesting and useful to investigate how the RFs can handle music speech and environmental sound discrimination tasks.

1.6 Thesis Outline

Following this introductory chapter, the rest of the thesis is structured as follows:

Chapter Two (Background and Current State of the Art): This chapter presents the state of the art research in the fields of audio content classification and automatic feature selection. The focus is on the used classification module, the classes to be detected, the used features, and the automatic feature selection technique where one is implemented.

Chapter Three (MPEG-7 Audio Features): This chapter discusses the features and their domains, and lists all of the features that are used in the training/classification process. Moreover, a mathematical definition is given for each feature.

Chapter Four (Machine Learning for Classification): This chapter reviews machine learning algorithms that are utilised in the proposed technique of audio content classification and feature selection. This includes GMM, NNet, decision tree and the ensemble classifier methods.

Chapter Five (Feature selection and Features Dimensionality Reduction): This chapter discusses the topics of feature selection and feature dimensionality reduction illustrating the difference between feature extraction and dimensionality reduction reviewing their common approaches. The second part discusses the proposed MPEG-7 feature dimensionality reduction techniques.

Chapter Six (Architecture of New High-Level Audio Content Classification and Feature Selection): This chapter describes the architecture and the design aspects of the proposed high-level audio classification system.

Chapter Seven (Audio Sample Database): This chapter discusses the available audio samples database and shows the need to create a new database one to test and evaluate the implemented classification system. The chapter also discusses the databases that are utilised for validation. Finally, some statistics of the utilised samples are listed.

Chapter Eight (Ensemble Tree Algorithms for High-Level Audio Content Classification and Feature Selection): This chapter proposes a systematic technique of the automated feature ranking and selection using ensemble trees, and illustrating the training and classification phases of the post-processing stage of the collective decision making.

Chapter Nine (Experiments and Results): This chapter contains the results of the conducted tests to show the performance of the proposed feature selection technique, and presents the resulting improvement in classification performance in testing and validation results.

Chapter Ten (Conclusion and Future Work): This chapter concludes the main contribution of this research and suggests the further work that can be performed in the future in order to achieve an automated metadata generation system.

Chapter 2: Background and Current State of the Art

2.1 Introduction

This chapter carries out a review of the current state of the art of the research in the field of audio content classification and feature selection techniques. The chapter comprises three main parts. The first part demonstrates the general structure of an audio content analysis. The second part covers the literature review of related publications about audio content classification system. The third part reviews the techniques used for feature selection in the field of audio content classification.

2.2 Audio Content Analysis

The topic of audio content analysis is a hot topic in the field of pattern recognition because audio is information rich media. The automation of audio content analysis will narrow the gap between human and machine, allowing more natural interaction with the digital devices and digital content.

Audio includes a wide range content such as speech, music and environmental sound. In the field of machine learning, some different techniques were developed over the years to address a specific task such as speech recognition, speaker identification, speech to text dictation, music information retrieval, query by example, query by humming, real time audio surveillance. These divert applications emphasise the importance of a pre-processing stage of identifying the general class of audio content in order to redeploy the content to a relevant information extraction technique.

2.3 MPEG-7 Hierarchical Audio Classification

There is a pressing need for an automated audio content analysis system. Such a system would enable automatic analysis and retrieval of the enormous amount of available digital audio content and automatically study audio file contents in order to generate a semantic text that describes the audio content efficiently. Such a system would help to avoid the need for manual indexing of audio files, which is especially important considering the enormous amounts of audio files which are available in archives, databases and the cloud. This would also contribute to the possible creation of an audio search engine that could work by query or by example.

One of the proposed applications of MPEG-7 is the multimedia mining system, which aims to retrieve audio information automatically by scanning multimedia data to detect the presence of specific contents similar to the content that the user has requested. Such a system contains two main components; the indexer and the multimedia mining server. The indexer produces an XML file and stores it in the multimedia mining server. Based on the indexing information, the multimedia mining server makes the required content available to the end user (Kim et al., 2005).

The MPEG-7 standard proposes a hierarchical audio classification system that contains three stages as in Figure 1. The first stage performs a high-level audio classification and segmentation. The second stage further processes the basic classes by performing a semantic classification. The third stage is the archiving stage, which utilises the storage and search of the index audio files (Kim et al., 2005).

This research will focus on the classification part of the proposed MPEG-7 system. Two aspects are studied; the first aspect is the classification and the second one is subset feature selection in order to improve the performance of the classification system.

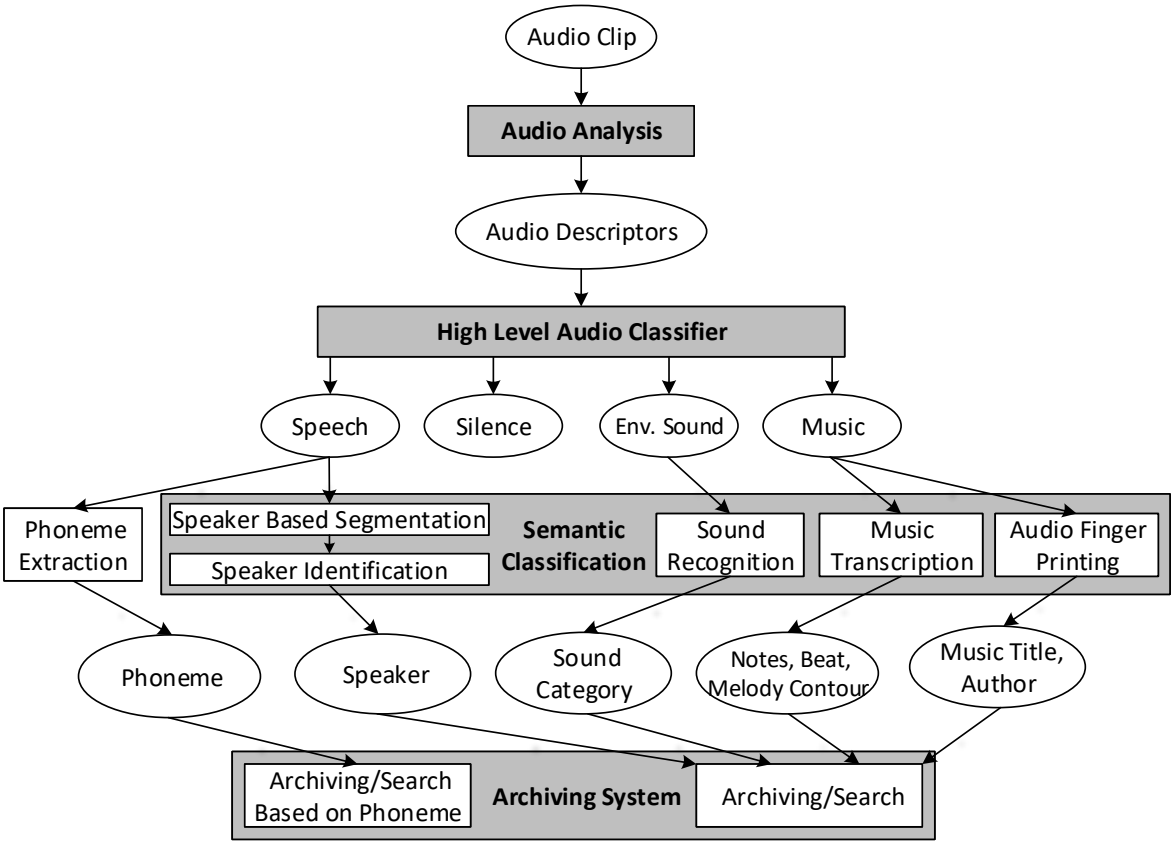


Figure 1: Audio content and scene analysis system architecture

2.4 Target Audio Classes

There is a wide variety of audible soundtracks and a successful audio classifier should be able to deal with these varieties efficiently. One approach to dealing with such a complicated classification task is to use a hieratical module that does the classification through multiple stages; the first stage performs a high-level audio classification, followed by subsequent levels of more detailed classification. The high-level audio classification classified the input audio into speech, music and environmental sound, these three classes were proposed by Gerhard (2003), followed by Temko (2007), Dennis (2014), and Al-Maathidi and Li (2015). Each one of the three aforementioned classes can be further classified to subcategories as shown in Figure 2 where small sample audio classes are illustrated.

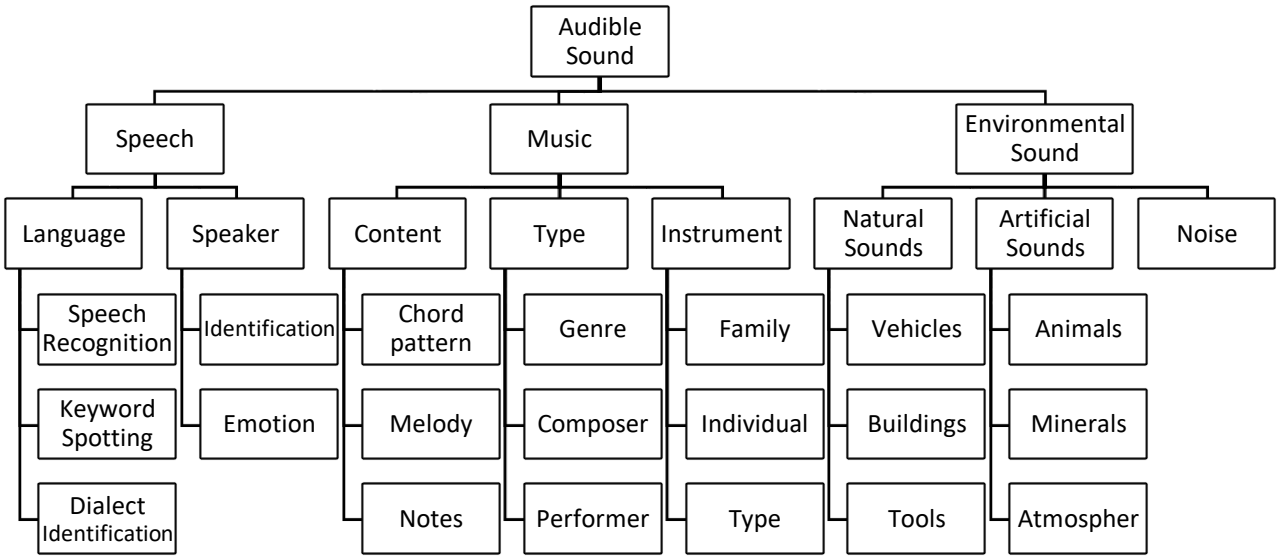


Figure 2: Audible sound classification (Gerhard 2003)

Such a high-level audio classification is vitally important as a pre-classifier for many other systems that are designed to deal with a content of a particular type to accomplish more specific tasks, the following publications are a sample of the available semantic classifiers that are designed to process the content of one of the three classes of speech, music and environmental sound:

1. Speech. There are plenty of publications deal with speech related topics. Some examples of which are speech detection (Bach et al., 2011), speech recognition (Pieraccini and Rabiner, 2012), and emotion recognition (Fewzee and Karray, 2013, Gharsalli et al., 2015), language identification (Muthusamy et al., 1994, Joshi and Joshi, 2013) and speaker clustering and identification (Liu et al., 2007, Shao et al., 2008), among others.
2. Music. In music, research has taken place in different areas. Some examples are genre classification (Tzanetakis and Cook, 2002; Yaslan and Cataltepe, 2006), geographical music origin identification (Zhou et al., 2014), instrument recognition (Morvidone et al., 2010), mood classification (Myint and Pwint, 2010; Mann et al., 2011; Li et al., 2015), room parameters acoustic parameters extraction (Kendrick et al., 2006), structure segmentation (Cheng et al., 2009), and many others.

3. Environmental sound. The environmental sound is the sound that does not contain speech or music data. Environmental sound can provide very important information for semantic analysis, there are many publications available in this field, for example the detection of bird species (Franzen and Gu, 2003; Jancovic and Kokuer, 2015), machine defect classification (Malhi and Gao, 2004), monitoring of human food intake (Kalantarian et al.; 2014, Bi et al.; 2016, Passler and Fischer, 2011), sport game segmentation (Zhang et al., 2010), violence detection (Jeho et al., 1998; Giannakopoulos et al., 2010, Acar et al., 2013), and many others.

The success of this application is highly dependent on a pre-processing stage of high-level audio classification that classifies audio content into one of these three classes to prepare each class content to be further processed by a proper semantic classifier. The aim of this research, is to propose a feature selection technique and test its performance on a high-level audio content classification system to classify the input audio content into speech, music and environmental sound. The next sections review the topics of audio content classification, feature dimensionality reduction and feature selection.

2.5 Audio Content Classification

The discipline of audio content analysis has been around for about 60 years. One of the earliest systems was built at Bell Labs by Davis et al. (1952). This system is able to recognise a single speaker spoken digits using spectral estimation using an analogue circuit. After the development of computers in the 1970s, many people entered the field of audio content classification. The available processing power and storage space limited the researchers to a basic set of features and simple classification methods but the rapid growth of computers processing power and the higher storage capacity enabled researchers to develop and utilise more sophisticated techniques.

The two major aspects of audio content classification system are the employed features and the utilised machine learning model. Each one of these two topics will be reviewed separately.

2.5.1 Audio Features for Classification

A large number of features were utilised in the field of audio content classification. These features can be categorised according to their domain. For example, the temporal domain, the frequency domain, the cepstrum domain, the modulation frequency domain, and the Eigen domain. A brief description of these domains is given (Salembier and Sikora, 2002; Mitrović et al, 2010; Tang et al, 2014; Alías et al, 2016):

- The temporal domain. The temporal domain is the native domain of audio signals, this domain contains a small set of features that can be easily calculated.
- The frequency domain. The features here are extracted from the frequency domain, this domain contains most of the audio features, audio data is presented in the frequency domain using transformation function such as Fourier transform, discrete cosine transform and wavelet transform. The reviewed features in this domain can be categorised into:
    - Physical features that describe the signal by its physical properties
    - Perceptual features that describe the signal by its perceptual properties.
- The cepstrum domain. The cestrum domain concept was introduced by Bogert et al. (1963), It is the result of taking the Fourier transform of the logarithm of the magnitude of the spectrum. The cepstrum features are considered a good technique for separating the components of a complex signal that are made up of several simultaneous but different elements combined together, such as speech features.

- Modulation frequency domain. The features in this domain are designed to capture a long-term signal variation of amplitude or frequency that is usually captured by a temporal analysis of the spectrogram.
- Eigen domain features. The features in this domain have a large amount of redundancy, therefore a statistical method is applied -Such as Principal Components Analysis (PCA) and Singular Value Decomposition (SVD)- to reduce the feature size while preserving most of the important information.

Table 2 lists a sample of features that covers all the aforementioned domains along with some basic features that are well established in the field. While Table 1 lists some abbreviations that are utilised in Table 2.

Table 1: Abbreviations utilised in Table 2

Column Title		Content	Meaning
Dimension	The dimensions of the feature	Number	Feature dimensions is equal to the mentioned number
		V	Variable
Application field	The field in which the feature was utilised	S	Automatic speech recognition
		E	Environmental sound recognition
		M	Music information retrieval
		G	Audio segmentation
		F	Audio fingerprinting
		V	Several applications
Computational complexity	Feature computational complexity	L	Low
		M	Medium
		H	High

Table 2: Audio Features

Domain	Feature Group	Feature Name	Perceptual	Psychoacoustic Models	Dimension	Application Field	Computational Complexity
Time (Temporal)	Zero Crossing	Zero Crossing Rate (ZCR)			1	V	L
		Linear Prediction Zero Crossing Rate			1	S	L
		Zero Crossing Peak Amplitudes		✓	V	S	M
		Pitch Synchronous Zero Crossing Peak Amplitudes		✓	V	S	M
	Amplitude	MPEG-7 Audio Waveform			2	V	L
		Amplitude Descriptor			9	E	L
	Power	Short-Time Energy, MPEG-7 Audio Power			1	V	L
		Volume			1	V	L
		MPEG-7 Temporal Centroid			1	M	L
		MPEG-7 Log Attack Time			1	M	L
Frequency (Perceptual)	Brightness	MPEG-7 Spectral Centroid	✓		1	M	L
		MPEG-7 Audio Spectrum Centroid	✓	✓	1	V	M
		Spectral Centroid	✓		1	V	L
		Sharpness	✓	✓	1	V	M
		Spectral Centre	✓		1	M	L
	Tonality	Bandwidth	✓		1	V	L
		MPEG-7 Audio Spectrum Spread	✓	✓	1	V	M
		Spectral Dispersion	✓		1	M	L
		Spectral Rolloff	✓		1	V	L
		Spectral Crest	✓		V	F	L
		Spectral Flatness	✓		V	F	M
		Subband Spectral Flux	✓		8	E	M
		(Multi-resolution) Entropy	✓		V	S	M
	Loudness	Sone	✓	✓	V	M	H
		Integral Loudness	✓	✓	1	M	H
	Pitch	Pitch (dominant frequency)	✓		1	V	L
		MPEG-7 Audio Fundamental Freq.	✓		2	V	L
		Pitch Histogram	✓		V	M	M
		Psychoacoustical Pitch	✓	✓	V	V	H
	Chroma	Chromagram	✓		12	M	M
		Chroma CENS Features	✓		12	M	M
		Pitch Profile	✓		12	M	H
	Harmonicity	MPEG-7 Audio Harmonicity	✓		2	V	M
		Harmonic Coefficient	✓		1	G	L
		Harmonic Prominence	✓		1	E	M
		Inharmonicity	✓		1	M	M
		MPEG-7 Harmonic Spectral Centroid	✓		1	M	M
		MPEG-7 Harmonic Spectral Deviation	✓		1	M	M
		MPEG-7 Harmonic Spectral Spread	✓		1	M	M
		MPEG-7 Harmonic Spectral variation	✓		1	M	M
		Harmonic Energy Entropy	✓		1	M	M
		Harmonic Concentration	✓		1	M	M
		Spectral Peak Structure	✓		1	M	M
		Harmonic Derivate	✓		V	M	M

Table 2: Audio features (continued)

Domain	Feature Group	Feature Name	Perceptual	Psychoacoustic Models	Dimension	Application Field	Computational Complexity
Frequency (Physical)	Auto Regression	Linear Predictive Coding			V	S	L
		Line Spectral Frequencies			V	V	M
	Adaptive Time Frequency Decomposition	Daubechies Wavelet Coefficients Histogram			28	M	M
		Adaptive Time-Frequency Transform			42	M	M
	Short-Time Fourier	Subband Energy Ratio			V	V	L
		Spectral Flux			1	V	L
		Spectral Slope			4	V	L
		Spectral Peaks			V	M	L
		Group Delay			V	S	M
Cepstral	Perceptual Filter Bank	Mel Frequency Cepstral Coefficients (MFCC)		✓	V	V	H
		Bark Frequency Cepstral Coefficients.		✓	V	V	H
		Autocorrelation MFCCs		✓	V	S	H
	Advanced Auditory Model	Noise-Robust Auditory Feature		✓	256	E	H
	Auto Regression	Perceptual Linear Prediction		✓	V	S	H
		Relative Spectral Perceptual Linear Prediction		✓	V	S	H
		Linear Prediction Cepstral Coef.			V	S	M
Modulation Frequency	Modulation Frequency Domain	Auditory Filter Bank Temp. Envelopes		✓	62	M	M
		Joint Acoustic and Modul. Freq. Feat.		✓	V	V	H
		4 Hz Modulation Harmonic Coef.			1	G	M
		4 Hz Modulation Energy		✓	1	G	M
	Rhythm	Band Periodicity	✓		4	G	M
		Pulse Metric	✓		1	G	M
		Beat Spectrum	✓		V	M	H
		Cyclic Beat Spectrum	✓		V	M	H
		Beat Tracker	✓		1	M	H
		Beat Histogram	✓		6	M	M
		DWPT-based Rhythm Feature	✓		V	M	M
		Rhythm Patterns		✓	80	M	H
Eigen		Rate-scale-frequency Features		✓	256	E	H
		MPEG-7 Audio Spectrum Bases			V	V	H
		Distortion Discriminant Analysis			64	F	H

After reviewing the features in Table 2, a selected sample of related publication is listed in Table 3, for each publication the table will illustrate the intended classification task and the utilised features.

Table 3: Classification tasks and the utilised audio features

Reference	Task	Utilised Features
(Kim et al., 2004)	Speaker and gender recognition.	MFCC, MPEG-7 spectral basis.
(Panagiotakis and Tziritas, 2005)	Silence, speech, and music.	RMS, and ZCR.
(Liu et al., 2007)	Classification and speaker identification	MFCC, sub-band energy ratio, ZCR
(Changseok et al., 2008)	Piano, guitar, flute, violin, and saxophone.	MFCC, and average MFCC
(Wang et al., 2008)	Music genre detection	MFCC, ZCR, spectral roll off, spectral centroid, and spectral flux
(Dhanalakshmi et al., 2009)	Music, news, ports, advertisements, cartoons, and movies	the Linear Predictive Coefficients, Linear Predictive Cepstral Coefficients, MFCC
(Dogan et al., 2009)	Audio segmentation to speech, commercials, environmental sound, silence	MPEG-7 features
(Kos et al., 2009)	Online speech/music segmentation	MFCC and MFCC-Variance
(Muhammad and Alghathbar, 2009)	Environment Recognition	MPEG-7 features
(Song et al., 2009)	Silence, pure speech, music, and non-pure speech	Short-time energy, zero-crossing rate, bandwidth, low short-time energy ratio, high zero-crossing rate, and noise rate
(Zhang et al., 2010)	Detect football, basketball, tennis, hockey, ping-pong, and badminton games	MFCC, perceptual linear predictive, short-time energy, spectrum flux, sub-band energy distribution, brightness and bandwidth
(Neal et al., 2011)	Bird species identification	normalised frame spectrogram
(Lampropoulos and Tsihrintzis, 2012)	Speech Emotional Recognition	MPEG-7 features
(Sonnleitner et al., 2012)	Speech Activity detection	Proposed a simple, efficiently computable spectral feature for precisely detecting spoken speech within complex, mixed audio streams. The RFs utilised for classification.
(Dadula and Dadios, 2015)	Detect abnormal event in public transport bus	MFCC

Reviewing the data presented in Table 2 and 3 emphasize the pressing need for an efficient feature selections technique that can suggest the optimum feature set for the individual classification task due to:

1. The huge number of available features, training the classifiers using a huge set of classification features increase the computational overhead, classification time and the probability of overfitting (were the classifier fit the training set very well but will fail to generalize to the new samples). This emphasizes the need for choosing a proper feature set for training.

2. There is no specific answer for the sufficient number of features that will work for the specific classification task.
3. Feature classification performance is not guaranteed for the intended classification task.
4. The performance of a specific feature is expected to vary according to the target classification class and the utilised training dataset.
5. Even for the well-established features like the MFCC, the classification performance can be improved by omitting some of the irrelevant features.

To be able to implement and evaluate MPEG-7 proposed audio content classification and feature selection technique, a set of features must be extracted first. Due to the nature of the diverse classes content -of speech, music, and environment sound- there is a need for a feature set that expected to capture the properties of these diverts classes successfully.

The proposed MPEG-7 audio features are a suitable choice due to the fact that they contain features that belong to multiple domains, with different complexity levels. The MPEG-7 audio features have some features that are designed to handle the content of each one of the target classes.

Therefore MPEG-7 audio features were chosen as the classification features to test and evaluate the implemented technique of feature selection and dimensionality reduction. MPEG-7 audio features have well defined mathematical definitions, it contains features that belong to multiple domains, with different complexity levels.

2.5.2 Machine Learning for Audio Content Classification

The selection of machine learning model can affect the performance of the classification system. There are several supervised machine learning models which can be adopted in this research.

Table 4 lists sample of related publications illustrating their intended classification task and the utilised classification module. Each row is dedicated to a single publication reference. The first column shows the reference, the second column illustrates the intended classification task and the reaming columns show the classification accuracy for the unlisted classification models.

Table 4: Machine learning for supervised audio content classification

Reference	Classification Task	Percentage of Positive Classification Accuracy						
		Decision Tree	Gaussian Mixture Model	Hidden Markov Model	*k*-Nearest Neighbors	Neural Network	Random Forests	Support Vector Machine
(Tzanetakis and Cook, 2002)	Music genre classification		61.0		60.0			
(Berenzweig et al., 2002)	Artist classification					64.9		
(Ziyou et al., 2003)	Sports audio classification			94.7				
(Ghaemmaghami, 2004)	Speech, music, speech/ music and others classification		90.5					
(Aronowitz, 2007)	Speech, music classification		98.6					98.2
(Changseok et al., 2008)	Content based audio classification	94.1			97.1			89.3
(Chu and Champagne, 2008)	Speech, music classification	84.8						86.0
(Wang et al., 2008)	Music genre classification					76.7	81.4	
(Dhanalakshmi et al., 2009)	Music, news, sports, advertisement, cartoon and movie classification					93.7		92.1
(Dhanalakshmi et al., 2010)	Music, news, sports, advertisement, cartoon and movie classification		92.9			93.1		
(Al-Maathidi and Li, 2012a)	Speech, music and event sound classification					91.3		86
(Yang and Su, 2012)	Audio event detection						76.6	
(Al-Maathidi and Li, 2015)	Speech, music and event sound classification		87.4			89.5	89.5	
(Murthy and Koolagud, 2015)	Song vocal and non-vocal regions identification					87.1		
(Zhang and Lv, 2015)	Environmental audio classification						96.2	

The data in Table 4 illustrate that GMM, NNet, RFs and SVM were the preferred option in the research community. Therefore, GMM, NNet, RFs will be adopted in testing and

evaluation in this study. SVM was omitted because it does not perform better than the other utilised techniques in the reviewed sample of papers, with the exception of one case, where it was compared with simple tree classifier and scored an improvement of 1.2% in classification accuracy by Chu and Champagne (2008), the performance of the RFs will definitely provide better improvement over the tree classifier. Furthermore, the SVM is reported to have a sensitivity for feature selection by Al-Maathidi and Li, (2012a) and sensitivity for kernel selection by Harrington (2012).

Including the RFs negates the need to include Decision Trees because it represents an improved version of decision tree classifier as will be illustrated later in Chapter 4. Finally, the hidden Markov model is not the preferred option for the task of audio classification using low-level descriptors, hidden Markov model strength lies in the ability to model the time structure, such as in work by Ziyou et al (2003) to classify sports audio content by searching a series of audio event and detect the specific sport. This also makes it one of the preferred options in speech recognition and many other similar application and not for frame by frame classification.

2.6 Feature Dimensionality Reduction in the Field of Audio Content Classification

Most research in the field of audio content classification focuses either on new features calculation or on employing different classification methods. On the other hand, less attention was given to features dimensionality reduction, even though it presented a critical step to improve the classification performance, especially when utilising high-dimensional feature space. Feature dimensionality reduction will improve classification accuracy, decrease the overfitting probability and reduce the training and classification time. The resulted optimised

features should have a high correlation to the target class and should be uncorrelated with each other (Guyon, 2003, Tang, 2014).

The two main approaches to feature dimensionality reduction are feature extraction and feature selection, Figure 3 lists sample methods of each approach.

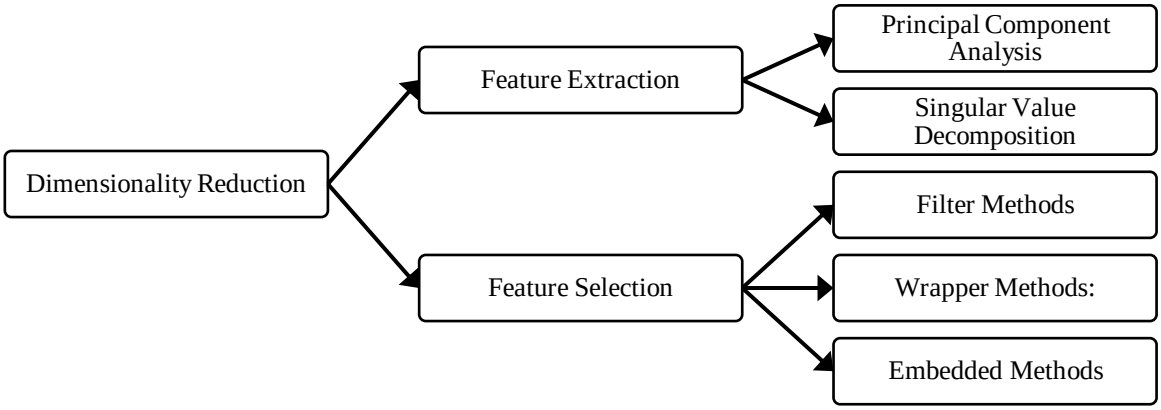


Figure 3: Feature dimensionality reduction

The key difference between the two approaches of dimensionality reduction is that the feature extraction can be considered as a function that maps the input features to a new set of feature set that has a lower dimensionality, while the feature selection performs a selection of a subset of the input features without manipulation their values. Both of these approaches are discussed in detail in Chapter 5.

Table 5 lists a sample of related papers illustrating the intended classification task and the utilised classification model. Each row is dedicated to a single publication, the first column shows the reference, the second illustrates the intended classification task, the third column shows the utilised feature dimensionality reduction technique and the reaming column show the classification accuracy for the unlisted classification models.

Table 5: Dimensionality reduction for audio classification

Reference	Classification Task	Approach	Utilised Technique	Brief Description
(Grimaldi et al., 2003)	Music classification	Selection/ extraction	Information gain, gain ratio / PCA	The performance of Information Gain, Gain Ratio and PCA was compared. Test results showed that PCA outperformed both of the information gain and gain ratio techniques, reporting less than 1% degradation in the performance even after reducing 80% of the utilised 25 features.
(Fiebrink and Fujinaga, 2006)	Music information retrieval	Selection (forward)/ extraction	PCA (wrapper)	The feature space included 74 low-level features. The results show that classification accuracy increased by 8.6% with feed forward feature selection and increase by 9.8% using PCA reduced features extraction.
(Rong et al., 2009)	Speech emotion recognition	Selection (ensembled)	Decision tree	The authors developed feature selection technique that is suitable to perform feature selection on a small data set that has a high number of features. The algorithm utilises both decision tree and RFs classifiers for feature selection. First, the multiple instances of random forests are trained using the original dataset, a result of bagging (discussed in Chapter 4), each tree in the forest will have a reduced instance of training features. The second stage is to generate a new enlarged dataset by combing multiple instances of forest feature for the class that positively by most of the trees in the forest. In the final step, decision tree is trained using enlarged features set, the features that score the highest majority voting is considered as a reduced feature set. The developed technique is suitable for the intended task of small data set with huge features, the novelty of this technique is about the use of RFs for feature set expansion to be able to use the decision tree for subset feature selection.

Table 5: Dimensionality reduction for audio classification (Continued)

Reference	Classification Task	Approach	Utilised Technique	Brief Description
(Yang and Su, 2012)	Auditory context classification	Selection	RFs	Three sets of classification features were utilised, including MFCC), hidden Markov model (HMM) audio event descriptor, and discrete wavelet packets descriptor (DWPD). The DWPD packets were reduced using the RFs classifier cation module was trained using 20 dimension DWPD features, 14 dimensions HMM features, and 21 dimensions MFCC based features. The experiment results show the effectiveness of the RFs classifier and the success of combining several heterogeneous features. The RFs was used twice, the first time for DWPD selection, and a second time in audio context classification. No feature selection step was introduced to reduce the 55 dimension features that include the DWPD, HMM, and MFCC features. The authors could have used the RFs after the extraction of all the features and not only on the DWPD features. The authors have not provided details about the DWPD feature selection.
(Räsänen and Pohjalainen, 2013)	Recognising the level of social conflict from speech	Selection (backwards)	*k*-NN classifier (Wrapper)	The proposed technique measure features using random subset feature selection, the quality of the individual feature is evaluated according to its contribution to the correct classifications. Feature selection is performed by two subsequent stages. The first stage computes the relevance of each feature using a fixed number of random subset classifiers and the second stage performing backwards elimination of the least relevant features.
(Baniya et al., 2014)	Music genre classification	Selection (forward)/ extraction	MRMR/PCA	The authors reported a classification accuracy of 87.9% using MRMR feature selection and 78.2% using PCA reduced features extraction after using only 37.17% of the original 538 feature.
(Delgado-Contreras et al., 2014b)	Environmental sound classification	Selection (filter)	Chi-square filter	The utilised fracture selection was able to reduce 24.19% of the input 62 features, the recorded class accuracy was dropped by 1.42% only after feature selection.
(Thambi et al., 2014)	Speech, non-speech detection	Selection	Subset Evaluation, correlation-based, gain-ratio, information-gain, relief, oneR and symmetric uncertainty	All the examined feature selection techniques gave a very comparable in performance, the best performance was achieved using correlation-based feature selection enabling features reduction from 272 to 8 features only. The RFs utilised for classification only and it scored the accuracy of 97.8% using the selected 8 features improving the result by 3.3% over the classification result using the whole 272 features.

In addition to the listed publications, the environmental audio data classification and feature selection were studied in Zhang and Lv (2015). The authors tested three tree ensemble methods including bagging, AdaBoost, and RFs. Two set of features were examined: MFCC features and linear predictive coefficient features.

The presented results show that RFs classification results outperform the other ensemble tree methods. Additionally, MFCC obtains a better classification result than the linear predictive coefficient based features. The feature importance was found by evaluation of the features GINI index in the trained RFs.

The presented feature selection is the closest technique found in the literature to the technique proposed in this thesis, but there are some major differences that include:

1. Different classification problems were addressed. In this thesis, the focus was to classify the audio into speech, music and environment sound, while Zhang and Lv (2015) work address environmental audio classification.
2. In this thesis, larger and more diverse set features were studied.
3. The work of Zhang and Lv (2015), the training/classification technique is different. This thesis utilises binary classifiers for each target class rather than a single multi-class classifier provide, the advantages of utilising multiple binary classifiers are discussed in Chapter 8.
4. As a result of having a distinct classification module for each target class, this allows the utilization of the post processing stage of collective decision making that can improve the classification results significantly.
5. The paper does not propose an automated technique for specifying the optimum size for the reduced feature space, as proposed in this research.

6. The technique proposed in this thesis allows the utilization of a subset of the class detectors -for example just a speech detector as a pre-processor for keyword spotting system-. As a result, the classification accuracy will be higher.

2.7 Other Related Feature Selection Papers

Feature selection has been utilised in many fields other than audio content classification. In the work of Peng et al. (2005) the author applied feature selection one multiple datasets. One dataset of handwritten digits, two other datasets of cancer types and a dataset of abnormal heart rhythm. The authors have done both forward and backwards feature selection using the minimum redundancy maximum relevance approach to rank the features according to their statistical properties. After feature ranking wrapper methods utilised in combination with three classifiers the Naive Bayes, Linear Discriminate Analysis and the SVM. The authors reported that through their comprehensive experiments on both discrete and continuous data sets and multiple types of classifiers demonstrate that the classification accuracy can be significantly improved based on minimum redundancy maximum relevance feature selection.

2.8 Relevant Publications on Ensemble Tree Feature Selection

The ensemble tree classification has been utilised in several fields of pattern recognition, but no publication has been found that suggests the use of ensemble for feature ranking and selection in the field of high-level audio content classification for the classes of speech, music and environmental sound. In our work of Al-Maathidi and Li (2015), the use of RFs and bagged tree (BT) for feature importance ranking and feature selection for high-level audio content classification was explored by showing how the RFs features ranking will affect the classification performance for the classes of speech, music and environmental sound.

The utilization of ensemble tree feature ranking and selection produces highly promising results in the field of high-level audio content classification, reported by the test results in Chapter 9 and in the research paper of Al-Maathidi and Li (2015), our test result indicate that reducing the classification features to about ten features cause only minor degradation in classification performance.

The following publications represent a selection of publications that utilise the RFs classifier for feature selection in some other fields of pattern recognition:

- Joelsson et al (2006) compare the performance of PCA reduced features extraction with RFs feature selection for morphological feature extraction for aerial images. It was found that RFs yields equal or better accuracies than PCA while using the same size if feature space.
- Reif et al. (2006) characterise the performance of RFs on genetic and proteomic datasets to identify the relevant features in high-dimensional data.
- Xiong and Wang (2009) propose a hybrid method based on improved ant colony optimisation and RFs for selecting a small set of marker genes from microarray data to produce a high-accuracy cancer classifier.
- Hu et al. (2009) used RFs for feature selection for intelligent disease and symptoms diagnosis modelling using five endogenous pathogens as a feature set.
- Wang et al. (2010) use an RFs data-mining method for feature selection for female sub-health state.
- Pang et al. (2012) use RFs for gene selection in high-dimensional data with survival outcomes.
- Kayim et al. (2013) use RFs for facial-feature selection for gender recognition. The results show that performance can be maintained while reducing the feature set significantly.

- Paja and Wrzesien (2013) use RFs to find the most important features that characterise melanocytic spots on the skin.
- Yamauchi (2013) studies computer mouse trajectories to find the user state anxiety; RFs was used to reduce 134 variables to select 3–8 key features.
- Guo et al. (2014) use RFs for feature selection in body-part recognition to estimate body posture.
- Gharsalli et al. (2015) use RFs to select important face-appearance features in the field of emotion recognition.
- Murata et al. (2015) use RFs to reduce the image feature vector in the field of image recognition.
- Uddin and Uddiny (2015) use RFs for feature selection, in the field of human activity recognition.
- Zhang and LV (2015) use RFs for feature selection, in the field of environmental audio classification.

As the literature shows, RFs is used successfully in many fields: Image pattern recognition; genetics; health care and disease diagnostics; emotion recognition and human activity recognition, none of these publications address the case of high level audio content classification, that emphasizes the need to pay this topic more attention.

2.9 Summary of Feature Selection

The following topics conclude the review of audio feature selection.

1. There are a vast amount of features available to summarise audio content and there is a huge amount of target classes that can be studied. However, most publications in the field of audio content classification either propose and discuss a specific classification technique or suggest the addition of new features to improve the performance.

2. There is a high demand for more efficient approaches for optimum feature selection especially in the field of audio content classification and retrieval.
3. Most of the utilised feature dimensionality reduction techniques does not take into consideration the feature contribution in classification. An example of this is the PCA feature extraction. The reduced feature set extracted by PCA relies on features covariance to perform the dimensionality reduction. This technique will work to some extent, but still there is a chance of losing some effective classification features that happen to have a lower variance, in this case, the PCA feature extraction will remove these effective features, and the classification performance will be decreased, as the result in Chapter 9 shows.
4. A better approach is to study the feature's behaviour inside an efficient and stable classifier to find the most contributing features, then use this knowledge to select optimal feature set.

Chapter 3: MPEG-7 Audio Features

3.1 Introduction

Feature extraction is an important step in audio classification systems. An efficient set of features should capture the most significant audio properties of the distinct classes to be analysed. This chapter starts with a preamble discussion of audio signal domains and implementation, lists the utilised audio features and illustrates their mathematical definitions, in order to enable the other researchers to reproduce the results and further develop it.

3.2 Audio Signal and Digital Representation

An audio signal can be described as a representation of a time variant sound pressure level. Audio signals can either be a periodic signal or a non- periodic signal, A periodic signal repeats itself over a constant time interval, while the non-periodic signal cannot be predicted even if the signal is observed over a period of time. The digitization of an audio signal is achieved by sampling the signal at specific times. The distance between each two samples is defined by equation 1:

$$T_s = \frac{1}{f_s} \qquad 1$$

where:

T_s is the time in seconds.

f_s is the sampling frequency.

After sampling, the highest frequency that can be reserved equal to half of the sampling rate, also referred to as the Nyquist frequency. Each sample is quantised using a pre-defined member of bits. The standard CD audio quality uses 16 bit sample depth and

44100 sampling rate. The audio samples used for training/evaluation saved using CD quality as discussed in Chapter 7.

The digital audio signal can be represented in more than one domain, A set of features can be extracted from each domain, such as time domain and the frequency domain which are used to extract the MPEG-7 features. Each of these two domains will be discussed briefly in the next sections.

3.2.1 Time Domain

The Time Domain, also known as the temporal domain, is the native domain for audio signals. In this domain, the signal is represented by its sampled amplitude over time. Therefore, the abscissa is time and the ordinate is amplitude. However, the features are basic in this domain as it will fail to differentiate the mixed-content audio samples. For example, the zero-crossing rate that has been mostly used to discriminate between speech and music will fail to discriminate between the mixed-type samples (speech with music background, and music mixed with environmental sound). It will also fail to discriminate between the genres of music (Panagiotakis and Tziritas, 2005).

3.2.2 Frequency Domain

In the frequency domain, the audio signal is represented by its spectral distribution which characterises the short-time spectrum. In the frequency domain, the abscissa is the frequency and the ordinates are the magnitude and phase.

To extract spectral features, the signal must be segmented into frames. The main reason for framing is that the audio signal is not stationary by nature.Therefore, windowing is introduced to make the signal look locally stationary, Windowing therefore utilises an efficient transformation of longer discrete signals into the frequency domain. The default

recommended window size in MPEG-7 standard is 30 ms, with a hop-size of 10 ms, the hop-size that represents the time interval between two successive frames (Kim et al, 2005).

Framing introduces discontinuities on the edges of the frame, these discontinuities will distort the resulted spectrum. In order to reduce this edge effect of framing the signal frame need to be multiplied by a windowing function There are many available windowing functions. For example, the Hamming window in equation 2 is one of these functions that is recommended by the MPEG-7 standard as a default windowing function. Figure 4 shows the Hamming window plot (Salembier and Sikora, 2002; Lerch, 2012; Giannakopoulos and Pikrakis, 2014).

$$w(n) = 0.54 - 0.64\cos\left(\frac{2\pi n}{N-1}\right), \qquad n = 0,1,...,N-1 \qquad 2$$

where N is the window size.

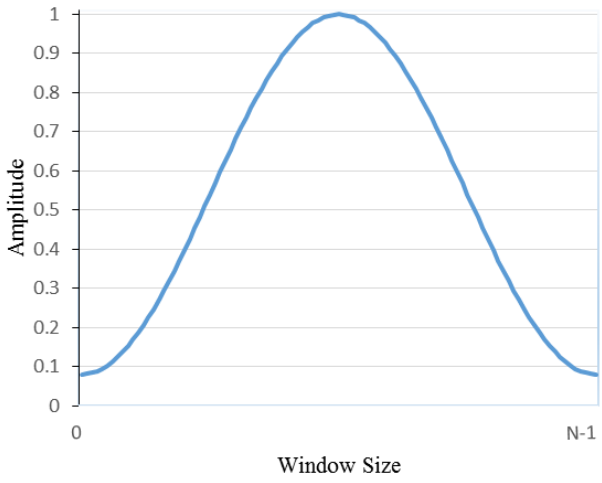


Figure 4: Hamming window

The resulted windows can be transformed into the frequency domain using the Discrete Fourier Transform (DFT) as equation 3 shows.

$$S_l(k) = \sum_{n=0}^{N_{FT}-1} s(n + lN_{hop})w(n)\exp^{\left(-i\frac{2\pi nk}{N_{FT}}\right)} \qquad (0 \le l \le L-1; 0 \le k \le N_{FT}-1) \qquad 3$$

where:

$S_l(k)$ is the l^{th} frame signal in frequency domain.

$s(n)$ is the signal in time domain.

N_{FT} is the size of Fast Fourier Transform.

N_{hop} is the integer number of time samples corresponding to hop-size.

L is the total number of frames in the signal.

$w(n)$ is the windowing function frequency bin index.

The average signal power is implementer in equation 4 according to Parseval's theorem.

$$\bar{P}_l = \frac{1}{N_{FT} E_w} \sum_{k=0}^{N_{FT}-1} |S_l(k)|^2 \qquad 4$$

where E_w is the window normalization factor that is defined in equation 5:

$$E_w = \sum_{n=0}^{N_w-1} |w(n)|^2 \qquad 5$$

The spectrum symmetry around Nyquist frequency allows to consider only the first half of the power spectrum without losing any information, so that $0 \le k \le N_{FT}/2$. To ensure that the summation of power coefficients is equal to the average power defined in equation 5, each power coefficient can be normalized using equation 6:

$$\begin{aligned} P_l(k) &= \frac{1}{N_{FT} E_w} |S_l(k)|^2 && \text{for } k = 0 \text{ and } k = \frac{N_{FT}}{2} \\ P_l(k) &= 2\frac{1}{N_{FT} E_w} |S_l(k)|^2 && \text{for } 0 < k < \frac{N_{FT}}{2} \end{aligned} \qquad 6$$

In the FFT spectrum the frequencies that correspond to bin index *k* are defined in equation 7 below that can be inverted to map the frequency in the range from 0 to $F_s / 2$ to a discrete bin from 0 to $N_{FT} / 2$ using equation 8 as follows:

$$f(k) = k \frac{F_s}{N_{FT}} \qquad (0 \le k \le N_{FT} / 2) \qquad 7$$

$$k = round\left(f \frac{N_{FT}}{F_s} \right) \qquad (0 \le f \le F_s / 2) \qquad 8$$

Figure 5 shows the spectrogram of a CD quality music audio signal, with N_{FT}=2048, hop-size=10 ms and frame size equal to 30 ms. A lighter shade indicates a higher power value.

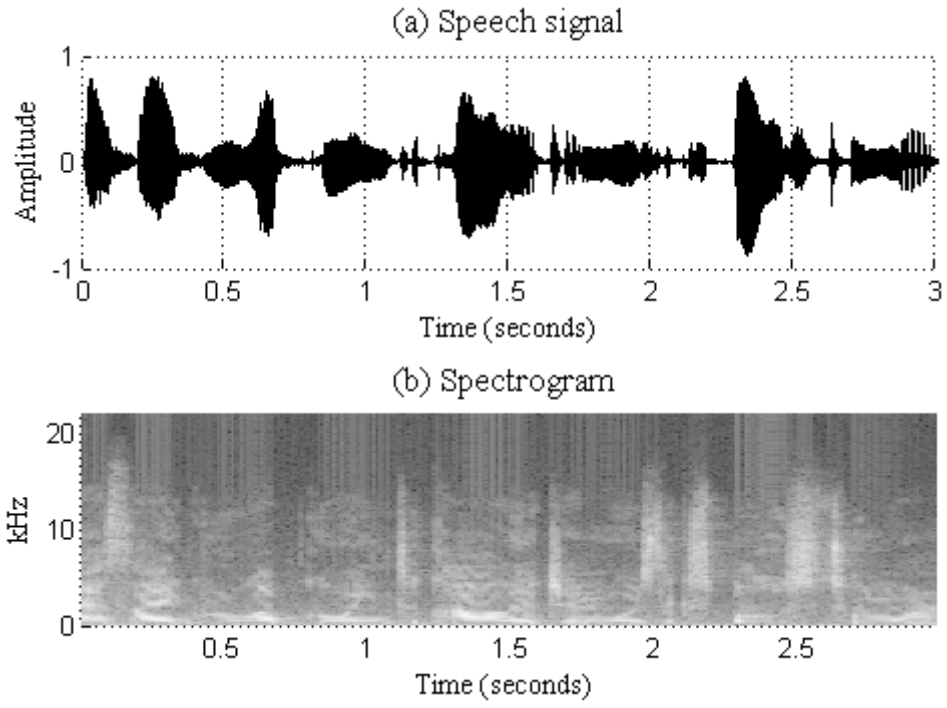


Figure 5: Spectrogram of music signal

3.3 MPEG-7 Low Level Descriptors for Audio

This research aims at studying automated feature ranking and feature selection. There are many features that can be tested, among which is the MPEG-7 LLD. These features were selected because they provide a simple and economical description of the temporal properties of the audio signal. They have also been used efficiently by many researchers in the field (Kim et al, 2004; Muhammad and Alghathbar, 2009; Lampropoulos and Tsihrintzis, 2012; Hossain and Muhammad, 2016). Part four of MPEG-7 standard gives the mathematical definition for a set of low-level audio descriptors, these descriptors have very general applicability in audio description. These descriptors can be grouped as shown in Figure 6 (Salembier and Sikora, 2002):

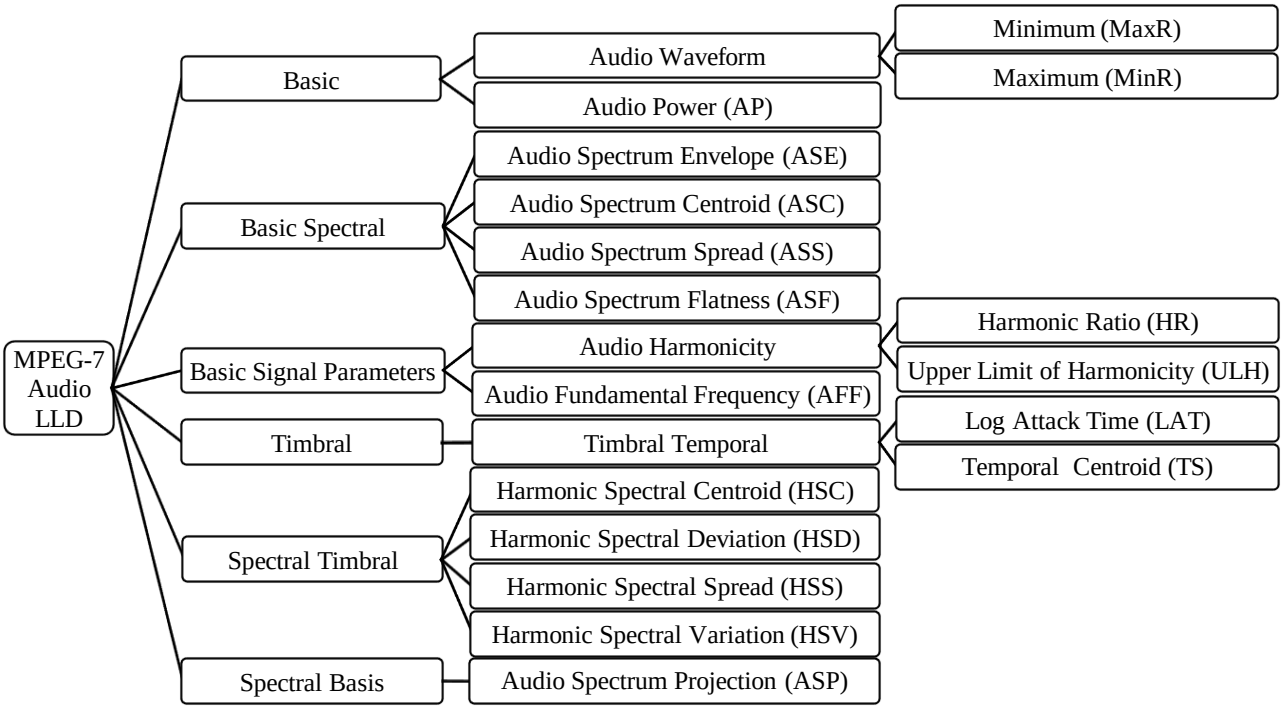


Figure 6: MPEG-7 low level audio descriptors

3.3.1 Basic Descriptors

The basic descriptors provide an easy to calculate features that describe the temporal properties of an audio signal.

3.3.1.1 Audio Waveform

The audio waveform (AWF) is an easy way to calculate feature that contains two values; the minimum range and the maximum range. The minimum range represents the lower limit of the frame amplitudes, while the maximum range represents the maximum limit of the frame amplitudes.

The AWF enables the estimation of the audio signal envelope in the time domain. Figure 7b gives graphical representations of the AWF descriptors.

3.3.1.2 Audio Power

Audio power (AP) describes the temporally smoothed instantaneous power of the audio signal. AP is calculated by finding the mean of squared frame samples of the audio file as equation 9 shows:

$$AP(l) = \frac{1}{N_{hop}} \sum_{n=0}^{N_{hop}-1} \left| s\left(n + lN_{hop}\right) \right|^2 \qquad \left(0 \le l \le L-1\right) \qquad 9$$

where

$s(n)$ is the audio signal.

l is the l^{th} frame in the signal.

N_{hop} is the hop size of the non-overlapping frames.

L is the total number of frames in the signal.

AP measures the amplitude evolution of the audio signal over the time. The combination of AP and some other basic MPEG-7 spectral descriptors can provide a quick representation of signal spectrogram. MPEG-7 proposes the use of non-overlapped windows. This will result having a feature vector of a length that is shorter than the all the other descriptors. Therefore, overlapped frames were utilized during the extraction of AP

feature in order to have a consistent length for all of the descriptors. Figure 7c shows that AP has power peaks when the original signal has a high amplitude:

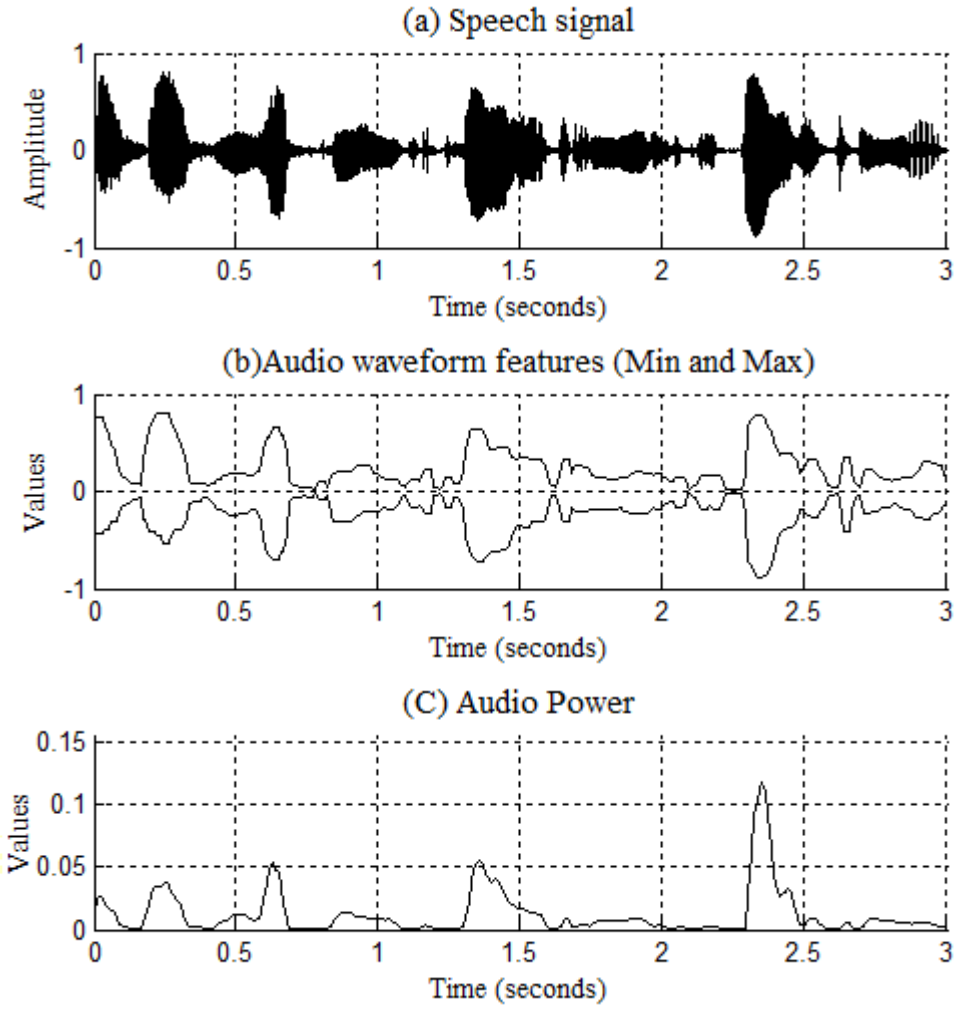


Figure 7: MPEG-7 basic descriptors extracted from music signal.

3.3.2 Basic Spectral Descriptors

The basic spectral descriptors contain four logarithmic frequency descriptors of the short-term audio power spectrum. The logarithmic scale of frequency is used to approximate the response of the human auditory system. These descriptors use the short-term power spectra of overlapping frames. For simplification purposes, the frame index *l* is omitted in the following equations.

3.3.2.1 Audio Spectrum Envelope

The audio spectrum envelope (ASE) describes the spectrum in logarithmic frequency scale, ASE can be used to approximate the reduced spectrogram of the audio signal. The ASE is calculated by finding the summation of the original power spectrum energy within a series of frequency bands.These bands are distributed logarithmically (base 2 logarithm). The edges of the frequency bands must be related 1 kHz using equation 10.

$$Edge = 2^{rn} \times 1\text{kHz} \qquad 10$$

where

r is the spectral resolution of the frequency in each band. It has the following eight possible values (1/16 octave, 1/8 octave, ¼ octave, …. 8 octaves).

n is an integer number.

The default value of the high edge is set to 16 kHz, the low edge is set to 62.5 and the default range is 8 octaves logarithmically centred at 1 kHz frequency.

The frequency edges can be defined using equation 11.

$$\begin{aligned} loF_b &= loEdge \times 2^{(b-1)r} \\ hiF_b &= loEdge \times 2^{br} \end{aligned} \quad \left(1 \le b \le 8/r\right) \qquad 11$$

where

b is the frequency band index.

r is the spectral resolution of the frequency in each band.

loF_b is the low frequency of the band *b*.

hiF_b is the high frequency of the band *b*.

loEdge is the lower edge of the band.

The values loF_b and hiF_b are rounded using equation 8 to get loK_b and hiK_b that will be used in the following equation.

Now the ASE can be defined by equation 12:

$$ASE(b) = \sum_{k=loK_b}^{hiK_b} P(k) \qquad (1 \le b \le 8/r) \qquad 12$$

where

b is the frequency band index.

loK_b is the low frequency of the band b.

hiK_b is the high frequency of the band b.

$P(k)$ is the power spectrum coefficients defined in equation 6.

Figure 8 shows ASE vector of 10 different values, 8 within the band coefficients and 2 out of the band coefficients. Figure 10b shows the ASE of music signal illustrated in Figure 10a.

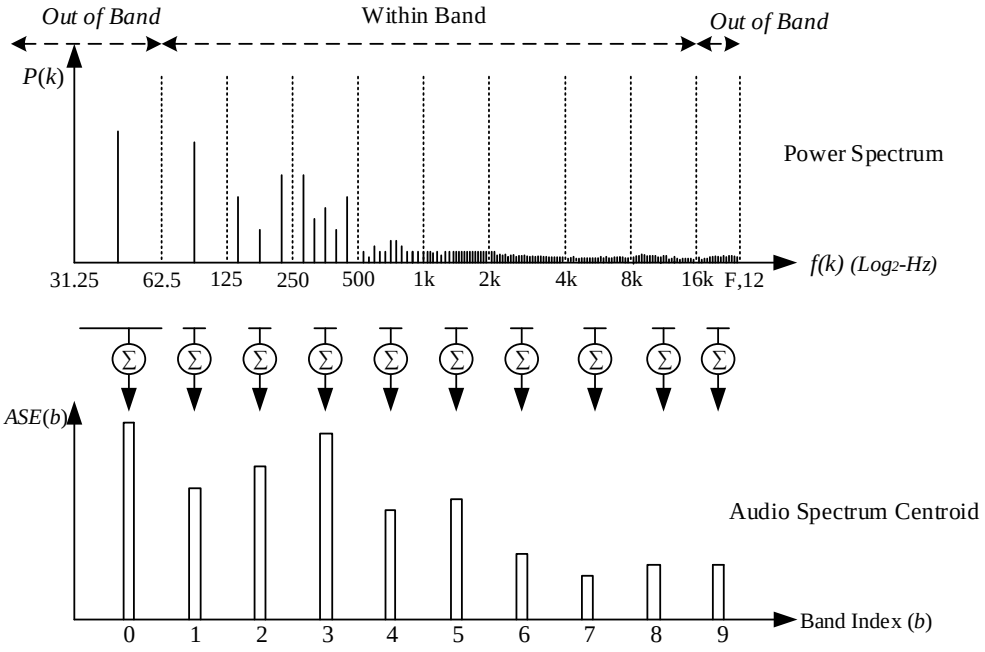


Figure 8: ASE extraction from power spectrum.

Figure 8 shows ASE vector of 10 different values: 8 within the band coefficients and 2 out of the band coefficients.

3.3.2.2 Audio Spectrum Centroid

The audio spectrum centroid (ASC) measures the centre of gravity of a log-frequency power spectrum. The summation of all power coefficients below 62.5 Hz is calculated in order to prevent disproportionate weighting for the DC component and the very-low-frequency components.

In the discrete frequency bins scale, the values are below the index defined by equation 13.

$$K_{low} = floor\left(62.5 / \Delta F\right) \qquad 13$$

where:

floor (x) is a function that returns the largest integer that is less than or equal to x.

ΔF is equal to Fs/N_{FT} which represents the frequency interval between two consecutive FFT bins.

The resulting power spectrum $P'(k')$ is related to the original spectrum $P(k)$ by the relation shown in equation 14.

$$P'\left(k'\right) = \begin{cases} \sum_{k=0}^{K_{low}} P\left(k\right) & \text{for } k' = 0 \\ P\left(k' + K_{low}\right) & \text{for } 1 \le k' \le \frac{N_{FT}}{2} - K_{low} \end{cases} \qquad 14$$

The frequencies $f'(k')$ that correspond to the new bins k' are given by equation 15.

$$f'(k') = \begin{cases} 31.25 & \text{for } k' = 0 \\ f\left(k' + K_{low}\right) & \text{for } 1 \le k' \le \frac{N_{FT}}{2} - K_{low} \end{cases} \qquad 15$$

where:

$f(k)$ is the discrete frequency corresponding to bin indexes k.

k is the nominal frequency of the low-frequency coefficient; it has been selected at the centre of the low-frequency band: $f'(0)$=31.25 Hz.

The ASC can then be defined by the relation between the modified power coefficients $P(k)$ and their corresponding frequencies $f(k)$, as shown in equation 16:

$$ASC = \frac{\sum_{k'=0}^{(N_{FT}/2)-K_{low}} log_2\left(\frac{f'(k')}{1000}\right) P'(k')}{\sum_{k'=0}^{(N_{FT}/2)-K_{low}} P'(k')} \qquad 16$$

Each modified power spectrum coefficient $f'(k')$ is weighted by the corresponding power coefficient $P'(k')$, and the log-frequency scaling is utilised to approximate the frequency perception of the human auditory system (Kim et al., 2005).

The ASC provides information on the power-spectrum shape: it indicates whether a power spectrum is dominated by high or low frequencies.It also provides an approximation of the perceptual sharpness of the signal. Figure 10c shows the ASC of a music signal.

3.3.2.3 Audio Spectrum Spread

The audio spectrum spread (ASS) measures the spectral shape; it has been defined in MPEG-7 standard as the second central moment of the log-frequency spectrum. ASS can be found by taking the root mean square deviation of the spectrum from its centroid ASC as in equation 17.

$$ASS = \sqrt{\frac{\sum_{k'=0}^{(N_{FT}/2)-K_{low}} \left[\log_2\left(\frac{f'(k')}{1000}\right) - ASC\right]^2 P'(k')}{\sum_{k=0}^{(N_{FT}/2)-K_{low}} P'(k')}} \qquad 17$$

where:

$P'(k')$ are the modified power spectrum coefficients from equation 14.

$f'(k')$ are the corresponding frequencies from equation 15.

N_{FT} is the size of the DFT.

The ASS indicates the distribution of the spectrum around its centroid. Thus, a low ASS value means that the spectrum power might be concentrated around the centroid, while a high value means that the spectrum power might be distributed across a wider range of frequencies. It is designed specifically to help differentiate between noise-like and tonal sounds (Kim et al., 2005). Figure 10c shows the ASS of music signal illustrated in Figure 10a.

3.3.2.4 Audio Spectrum Flatness

The audio spectrum flatness (ASF) describes the power spectrum flatness of a given audio signal frame. ASF is represented by a series of values in which each value describes the spectrum flatness of a predefined band by finding the power spectrum deviation from a flat shape. For example, it can indicate similarity between the audio signal and the white noise.

To extract the ASF, the signal frame power spectrum is calculated first by using equation 6 and a non-overlapping frame of size recommended to be a 30ms.

The spectrum then is divided into ¼-octve spaced log frequency bands, so the low edge and high edge is calculated using equation 18.

$$loEdge = 2^{\frac{1}{4}n} \times 1\text{kHz}$$
$$hiEdge = 2^{\frac{1}{4}B} \times loEdge \quad (18)$$

where

loEdge is the lower edge of the band.

hiEdge is the higher edge of the band.

n is used to determine the lower band edge, it is recommended to set to n=8 so the *loEdge* become equal to 250 Hz.

B is used to determine the higher band edge. It should be set to a value so that the *hiEdge* does not exceed any of Nyquist frequency or the original signal bandwidth.

ASE feature is too sensitive to the variation in sampling frequency. Therefore, the *loEdge* and *hiEdge* frequencies slightly overlap, so that each frequency band is made 10% larger as equation 19 shows (Kim et al., 2005).

$$\begin{aligned} loF_b &= 0.95 \times loEdge \times 2^{\frac{1}{4}(b-1)} \\ hiF_b &= 1.05 \times loEdge \times 2^{\frac{1}{4}b} \end{aligned} \qquad (1 \le b \le B) \qquad 19$$

where

b is the b^{th} frequency band.

B is the number of frequency bands.

loF_b is the lower limit of band *b*.

hiF_b is the upper limit of band *b*.

loEdge is the lower edge of the band.

hiEdge is the higher edge of the band.

n is used to determine the lower band edge. It is recommended to set to n=-8 so the *loEdge* become equal to 250 Hz.

The values loF_b and hiF_b is rounded using equation 6 to get loK_b and hiK_b that will be used in equation 20.

MPEG-7 standard groups the power spectrum coefficients $P(k)$ in order to reduce computational costs using the following steps:

- For all bands between 1 kHz and 2 kHz, the power spectrum coefficients P(k) are grouped by pairs by averaging each two successive power spectrum coefficients.
- Within all bands between 2^n kHz and 2^{n+1} kHz (where n is an integer that is greater or equal to one), each group of 2^{n+1} successive power coefficients is replaced by a single coefficient equal to their arithmetic mean.
- The last group at end of each band may not contain the required number of coefficients. Therefore, if 50% or more of the required coefficients are missing the last group will be ignored. Otherwise, the reaming of the required coefficients is taken from the beginning of the next band and the arithmetic average is calculated.

After grouping the power, coefficients are denoted as $P_g(k')$, and the new band edges is denoted as loK_b' and hiK_b'. The grouping results are illustrated in Figure 9.

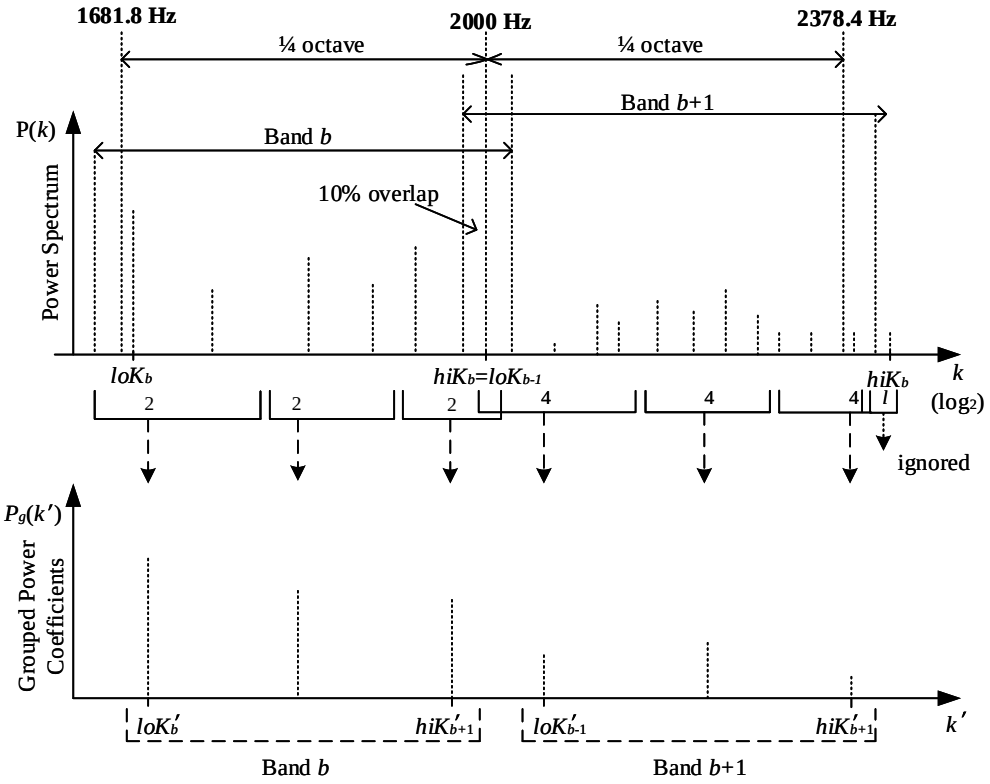


Figure 9: Grouping of power coefficients within 2 bands around 2kHz.

For each band, the ASF is estimated as the ratio between the geometric mean and the arithmetic mean of the spectral power coefficients within the band using equation 20.

$$ASF(b) = \frac{\sqrt[hiK'_b - loK'_b + 1]{\prod_{k'=loK'_b}^{hiK'_b} P_g(k')}}{\frac{1}{hiK'_b - loK'_b + 1} \sum_{k'=loK'_b}^{hiK'_b} P_g(k')} \qquad (1 \le b \le B) \qquad 20$$

If the result shows a flat ASF spectrum shape, this indicates that the input signal is a noise or an impulse signal. The high ASF coefficients are expected to reflect noisiness, while the low ASF values may indicate a harmonic structure of the spectrum. From a psychoacoustical point of view, a large deviation from a flat shape generally characterizes tonal sounds.

The ASF vector can be averaged to reduce the spectral flatness features to a single scalar that represents an overall flatness of a frame (Burred and Lerch, 2003; Burred and

Lerch, 2004). The average ASF will be utilised as a classification feature during feature extraction. Figure 10c shows the ASF of a music signal.

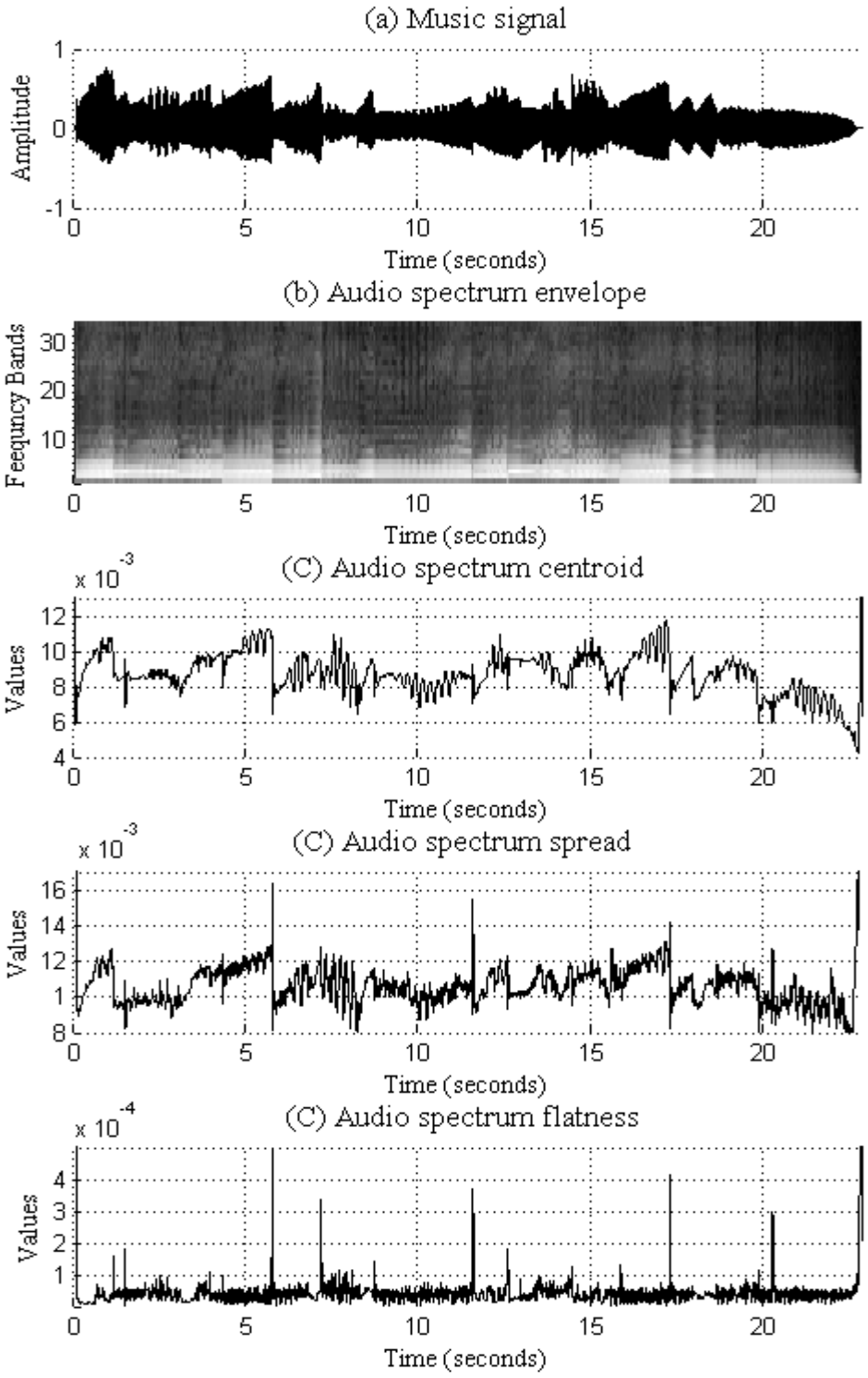


Figure 10: MPEG7 basic spectral descriptors extracted from music signal.

3.3.3 Basic Signal Parameters

The aforementioned basic spectral descriptors can give a smoothed representation of power spectra. However, they cannot describe the harmonic structure of periodic sounds in detail due to the lack of frequency resolution. The following descriptors provide a complementary information to describe harmonicity degree of audio signals:

3.3.3.1 Audio Harmonicity

The audio harmonicity (AH) designed to provide a compact description of the harmonic properties of sounds; the first is the harmonic ratio (HR) and the second is the upper limit of harmonicity (ULH). These measurements rely on a standardized fundamental frequency estimation method, based on the local normalized autocorrelation function of the signal. This approach is widely used for local pitch estimation.

These descriptors can be used to distinguish between the harmonic sounds, musical instruments sound and voiced speech segments for example, and non-harmonic sounds such as noisy sounds and unvoiced speech segments. The definition of these two descriptors is discussed in the subsequent sections.

Harmonic Ratio

The harmonic ratio (HR) measures the proportion of harmonic components in the power spectrum. HR is calculated for overlapped frames of the audio signal. The extraction of the HR is standardised by the MPEG-7 as follows:

First, the normalized autocorrelation function of the signal is estimated using equation 21:

$$\Gamma_l(m) = \frac{\sum_{n=0}^{N_w-1} s_l(n)\, s_l(n-m)}{\sqrt{\sum_{n=0}^{N_w-1} s_l(n)^2 \sum_{n=0}^{N_w-1} s_l(n-m)^2}} \qquad (l \le m \le M; 0 \le l \le L-1) \qquad 21$$

where

l is the frame index.

L is the total number of frames in the signal $s(n)$.

$s_l(n)$ is defined as $s_l(lN_{hop}+n)$ and N_{hop} is the hop between successive frames.

m is the lag index of correlation.

M is the maximum fundamental period that is equivalent to the minimum fundamental frequency which can be estimated by using equation 22.

$$M = T_0^{max} F_s = \frac{F_s}{f_0^{min}} \qquad 22$$

T_0^{max} here is the default maximum period and is equal to 40ms. That corresponds to f_0^{min} 25Hz of minimum fundamental frequency.

A purely periodic signal will have the maximum values of $\Gamma_l(m)$ at lags *m*, corresponding to multiples of T_0. For nearly any audio signal, despite its degree of periodicity, a peak with a value close to one probably will appear near lags with *m* = 0.

To obtain the HR, the maximum autocorrelation is found after ignoring the zero-lag peak using equation 23.

$$HR = \max_{M_0 \le m \le M} \{\Gamma_l(m)\} \qquad 23$$

where M_0 is the lag directly on the right of the zero-lag peak.

The MPEG-7 modifies equation 23 and writes it as shown in equation 24:

$$HR = \max_{l \le m \le N_{hop}} \left\{ \Gamma_l(m) \right\} \qquad 24$$

The difference in equation 24 is that the zero-lag peak is not ignored, which would result in HR values virtually always close to one. Also, the rightmost limit corresponds only to a frame length is not the maximum lag *M*. The lag that maximizes $\Gamma_l(m)$ corresponds to the estimated local fundamental period. The HR values will be close to zero for white noise and to one for purely periodic signals. Figure 11 shows the HR values extracted from three different audio signals that contain the sounds of a flute laughter and noise.

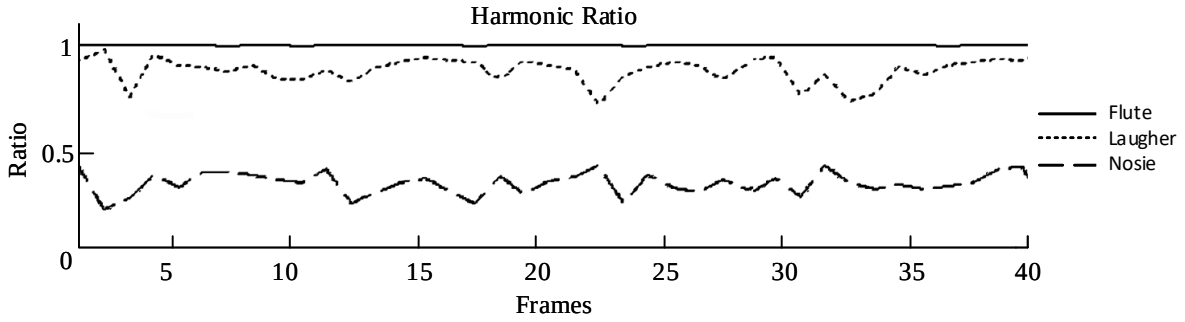


Figure 11: MPEG-7 HR and modified HR features extracted from three audio signals.

Upper Limit of Harmonicity

The upper limit of harmonicity (ULH) is an estimation of the frequency above which there is no harmonic structure. The algorithm of extracting ULH is based on the output/input power ratio of a time domain comb filter. The extraction is performed as follows (Moorer, 1974):

1. The comb-filtered signal is calculated using equation 25:

$$\tilde{s}_l(n) = s_l(n) - G_l s_l\left(n - \Gamma_l(m)\right) \qquad \left(0 \le n \le N_w - 1\right), \qquad 25$$

where G_1 is the optimal gain of the comb filter that is defined by equation 26:

$$G_l = \frac{\sum_{j=0}^{N_w-1} s_l(j)\, s_l(j-\hat{m})}{\sum_{j=0}^{N_w-1} s_l(j-\hat{m})^2} \qquad 26$$

2. The power spectra of the original signal $P'(k)$ and comb-filtered signals $P'_c(k)$ for each frame l are calculated using equation 14.

3. The ratio of $P'_c(k)$ summation to $P'(k)$ summation is calculated. The summation is calculated for power samples that are beyond a given frequency limit, as equation 27 illustrates.

$$R(k_{lim}) = \frac{\sum_{k=k_{lim}}^{(N_{FT}/2)-K_{low}} P'_c(k)}{\sum_{k=k_{lim}}^{(N_{FT}/2)-K_{low}} P'(k)} \qquad 27$$

where k_{lim} is the given frequency limit:

4. The ratios $R(k_{lim})$ are computed for k_{lim} from $k_{lim} = k$ down to the first frequency bin k_{ulh} in which $R(k_{lim})$ falls below the threshold of 0.5.

5. The corresponding frequency f_{ulh} is given the value of 31.25Hz if f_{ulh} is equal to zero, otherwise f_{ulh} will be given the value of $f(k_{ulh} + K_{low})$ as defined in equation 16.

6. Finally, the ULH feature is computed for each frame in the signal using equation 28:

$$ULH = \log_2\left(\frac{f_{ulh}}{1000}\right) \qquad 28$$

3.3.3.2 Audio Fundamental Frequency

The audio fundamental frequency (AFF) descriptor gives an estimate to the fundamental frequency f_0 in segments of a signal that is assumed to be periodic. AFF is

useful to get an approximation of the pitch of the given audio signal. It is mainly used to estimate the pitch of musical sounds and voiced speech.

There are many algorithms in the literature to estimate AFF. MPEG-7 standard has not specified any normative extraction method; therefore, the common approach of temporal autocorrelation is adopted to estimate the AFF is the temporal autocorrelation method described in previous section equations 21 and 23. The AFF used mainly to estimate the pitch of voiced speech and musical sounds. The pitch curve of a speech signal reflects the voice intonation and is an important prosodic feature. Figure 12b shows the AFF feature that is extracted from music signal in Figure 12a.

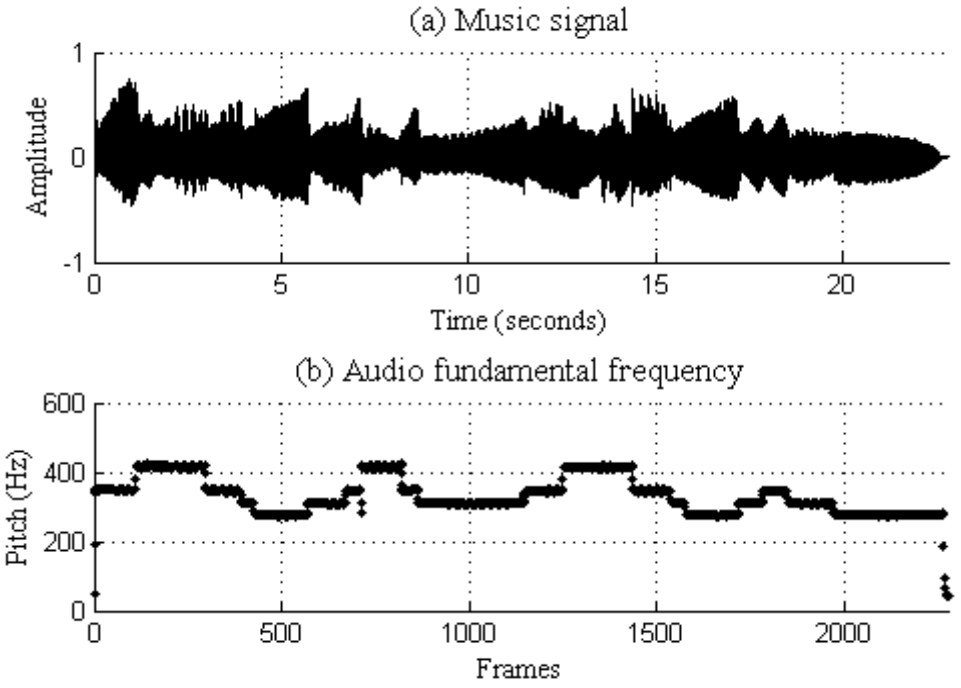


Figure 12: MPEG-7 AFF extracted for a music signals.

3.3.4 Timbral Descriptors

The term timbre refers to the features that enable differentiating between two sounds that are equal in pitch, loudness and subjective duration. It is used to describe the features of instrument sounds. There are two types of MPEG-7 timbre features:

1. The temporal timbral descriptors: they include the log attack time and temporal centroid.
2. The spectral timbral descriptors: they include the harmonic spectral centroid, harmonic spectral deviation, harmonic spectral spread, harmonic spectral variation and spectral centroid.

The following sections will discuss the calculation of these features in detail.

3.3.4.1 The Temporal Timbral Descriptors

The temporal timbral descriptors are extracted from the signal envelope. The signal envelope describes the energy change of the signal in the time domain. It is generally equivalent to the attack, decay, sustain, and release (ADSR) phases of a sound:

- *Attack* is the time taken for the sound to reach its initial maximum volume.
- *Decay* is the time taken for the sound to reach a second volume level known as the sustain level.
- *Sustain* is the volume level the sound sustains after the decay phase.
- *Release* is the time taken to reduce the volume to zero level.

These four phases of a musical sound are illustrated in Figure 13.

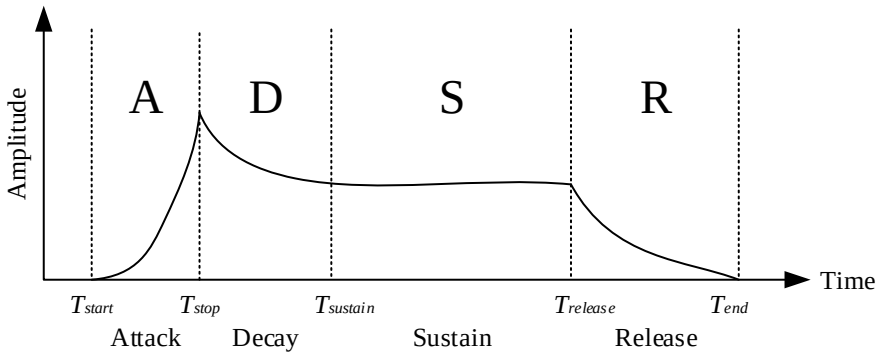


Figure 13: ADSR envelop general shape

One way to calculate the signal envelope is by Finding the RMS of the original signal frame by frame using equation 29:

$$Env(l) = \sqrt{\frac{1}{N_w}\sum_{n=0}^{N_w-1} s^2\left(lN_{hop}+n\right)} \qquad (0 \le l \le L-1) \qquad 29$$

where

$Env(l)$ is the envelope of frame l.

n is the index of a signal.

$S(n)$ is the original signal at time sample.

N_{hop} is the number of samples between two successive frames.

N_w is the number of frames in the time samples.

L is the total number of frames.

Log Attack Time

The log attack time (LAT) is the log of the time that the signal takes to reach the maximum amplitude of a signal from a minimum threshold time (Mcadams, 1999). LAT aims to describe the onsets of single sound samples for different musical instruments. equation 30 is used to find the LAT:

$$LAT = \log_{10}\left(T_{stop} - T_{start}\right) \qquad 30$$

where

T_{start} is the signal starting time.

T_{stop} is the time in which the signal reaches its maximum value if it has a decay phase, or it sustained part if it does not have a decay phase.

The MPEG-7 standard has not specified the method of determining the exact value of T_{start} and T_{stop}. The value of T_{start} is therefore assigned to the time that the signal envelope

takes to exceed 2% of its maximum value, whereas T_{stop} is assigned to the time that the signal envelope takes to reaches its maximum value (Kim et al., 2005).

Temporal Centroid

The temporal centroid (TC) is defined as the time average over the energy envelope of the signal. The value of TC is measured in seconds and calculated using equation 31:

$$TC = \frac{N_{hop}}{F_s} \frac{\sum_{l=0}^{L-1} \left(lEnv(l) \right)}{\sum_{l=0}^{L-1} Env(l)} \qquad 31$$

where

TC is the Temporal Centroid that is measured in seconds.

$Env(l)$ is the signal envelope defined in equation 29.

N_{hop} is the number of samples between two successive frames.

F_s is the sampling frequency.

The factor (N_{hop}/F_s) is used to convert the discrete frame index to the continuous time domain. Figure 14 illustrates the extracted value of LAT and TC from a dog bark sound.

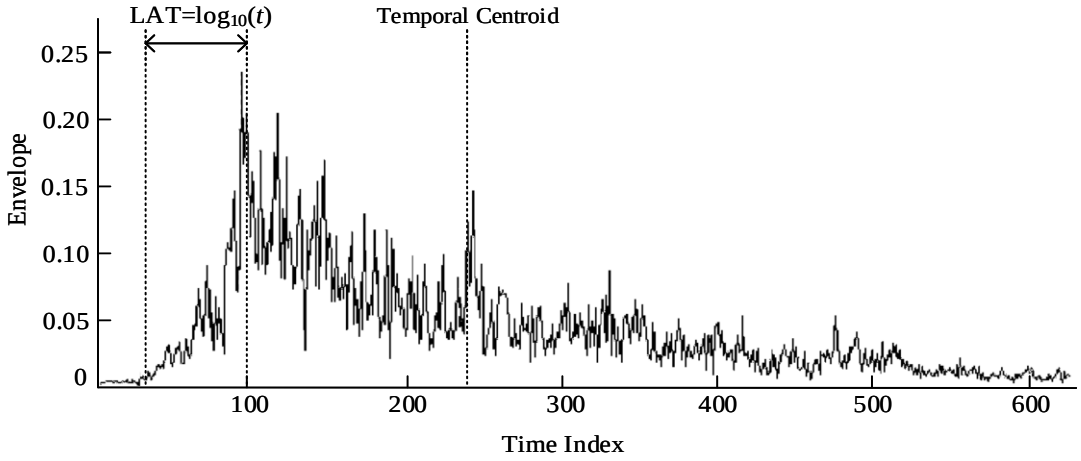


Figure 14: The temporal timbral descriptors extracted from a dog bark sound.

3.3.4.2 Spectral Timbral Descriptors

Spectral timbral features provide a description of the structure of harmonic spectra. These features are extracted in a linear frequency space. These features are designed to be applied on frames to get instantaneous values. Moreover, it can be applied to larger analysis windows to provide a global descriptor. In this thesis, only the frame level features will be discussed in order to comply with the rest of the LLD classification and feature selection.

MPEG-7 standard recommends the use of a Hamming window of size 30ms with 10ms hop size.

The fundamental frequency and the harmonic components of the signal need to be extracted first to enable the extraction of the spectral timbral descriptors. To detect the pitch and the harmonic peaks, the following four steps are carried out:

1. Calculating of the spectrum windowed signal $S(k)$ using equation 3, then the amplitude spectrum $|S(k)|$ is computed.

2. Calculating the fundamental frequency using equation 23.

3. Detecting spectrum peaks.

4. Analysing the candidate to determine if the peaks are a harmonic peak or not. Harmonic peaks are located around the multiples of the fundamental frequency.

3.3.4.3 Harmonic Spectral Centroid

The harmonic spectral centroid (HSC) is the average of the amplitude-weighted mean of the spectrum harmonic peaks. First, the frame level local harmonic spectral centroid (LHSC) is computed using equation 32:

$$LHSC_l = \frac{\sum_{h=1}^{N_H}\left(f_{h,l}A_{h,l}\right)}{\sum_{h=1}^{N_H}A_{h,l}} \qquad 32$$

where

l is index of the given frame.

$f_{h,l}$ is the frequency of the h^{th} harmonic peak in the l^{th} frame.

$A_{h,l}$ is the amplitude of the h^{th} harmonic peak in the l^{th} frame.

N_H is the number of harmonics that is taken into account.

Then HSC value is obtained by averaging the LHSC using equation 33:

$$HSC = \frac{1}{L}\sum_{l=0}^{L-1}LHSC_l \qquad 33$$

where L is the number of frames in the sound signal.

3.3.4.4 Harmonic Spectral Deviation

Harmonic spectral deviation (HSD) measures the deviation of the harmonic peaks from the envelopes of the local spectra. Before computing HSD, the spectral envelope is estimated by interpolating the adjacent harmonic peak amplitudes as equation 34 illustrates:

$$SE_{h.l} = \begin{cases} 1/2\left(A_{h,l} + A_{h+1,l}\right) & \text{if h=1} \\ 1/3\left(A_{h-1,l} + A_{h,l} + A_{h+1,l}\right) & \text{if } 2 \le h \le N_H - 1 \\ 1/2\left(A_{h-1,l} + A_{h,l}\right) & \text{if } h=N_H \end{cases} \qquad 34$$

where

$SE_{h,l}$ is spectral envelop of the frequency of the h^{th} harmonic peak in the l^{th} frame.

$A_{h,l}$ is the amplitude of the h^{th} harmonic peak in the l^{th} frame.

After that, the local harmonic spectral deviation (LHSD) is computed on a frame level using equation 35:

$$LHSD_l = \frac{\sum_{h=1}^{N_H} \left| \log_{10}\left(A_{h,l}\right) - \log_{10}\left(SE_{h,l}\right) \right|}{\sum_{h=1}^{N_H} \log_{10}\left(A_{h,l}\right)} \qquad 35$$

where

l is the detected harmonic peaks.

N_H is the amplitude of the h^{th} harmonic peak in the l^{th} frame.

$A_{h,l}$ is the amplitude of the h^{th} harmonic peak in the l^{th} frame.

$$HSD = \frac{1}{L}\sum_{l=0}^{L-1} LHSD_l \qquad 36$$

where L is the number of frames in the sound signal.

3.3.4.5 Harmonic Spectral Spread

Harmonic spectral spread (HSS) measures the average spectrum spread in relation to the HSC. The frame level HSS is defined as the power-weighted RMS deviation from the local HSC. Before computing HSS the frame level, local harmonic spectral spread (LHSS) is computed using equation 37:

$$LHSS_l = \frac{1}{LHSC_l}\sqrt{\frac{\sum_{h=1}^{N_H}\left[\left(f_{h,l} - LHSC_l\right)^2 A_{h,l}^2\right]}{\sum_{h=1}^{N_H} A_{h,l}^2}} \qquad 37$$

where

l is the index of the given frame.

$f_{h,l}$ is the frequency of the h^{th} harmonic peak in the l^{th} frame.

$A_{h,l}$ is the amplitude of the h^{th} harmonic peak in the l^{th} frame.

N_H is the number of harmonics that are taken into account.

Then HSS value is obtained by averaging the LHSS using equation 38:

$$HSS = \frac{1}{L}\sum_{l=0}^{L-1} LHSS_l \qquad 38$$

where L is the number of frames in the sound signal.

3.3.4.5 Harmonic Spectral Variation

The harmonic spectral variation (HSV) measures the spectral variation between adjacent frames. Before computing HSV the frame level local harmonic spectral variation (LHSV) is defined using the complement to one of the normalized correlation between the amplitudes of harmonic peaks of two adjacent frames as Equation 39 illustrate.

$$LHSV_l = 1 - \frac{\sum_{h=1}^{N_H}\left(A_{h,l-1}A_{h,l}\right)}{\sqrt{\sum_{h=1}^{N_H}A_{h,l-1}^2}\sqrt{\sum_{h=1}^{N_H}A_{h,l}^2}} \qquad 39$$

where

l is the index of the given frame.

$A_{h,l}$ is the amplitude of the h^{th} harmonic peak in the l^{th} frame.

N_H is the number of harmonics that are taken into account.

Then HSV value is obtained by averaging the LHSV using equation 40:

$$HSV = \frac{1}{L}\sum_{l=0}^{L-1} LHSV_l \qquad 40$$

where L is the number of frames in the sound signal.

3.3.5 Spectral Basis

The goal spectral basis descriptor is to provide a reduced dimension representation of the signal audio spectrum. This will improve the classification systems efficiency. As a starting point, the normalized audio spectrum envelop (NASE) is calculated. This is followed by a dimensionality reduction technique such as the PCA and SVD. These dimensionality reduction techniques will be discussed in detail in Chapter 5.

3.3.5.1 Audio Spectrum Projection

The audio spectrum projection (ASP) feature is found by multiplying the NASE matrix with basis function to achieve the required dimensionality reduction.

The following steps are carried out to calculate the NASE matrix:

1. Finding the audio spectrum envelop (ASE) matrix for the input audio signal *s* (*n*) using overlapped frames by applying equation 12.
2. The resulted log-frequency power spectrum is converted from logarithmic scale to decibel scale using equation 41:

$$ASE_{dB}(l, f) = 10 \log_{10}\left(ASE(l, f)\right) \qquad 41$$

where

l is the frame index.

f is the index of an ASE frequency.

3. Normalising the decibel-scale spectral vector with RMS energy envelope using equation 42:

$$X(l,f)=\frac{ASE_{dB}(l,f)}{\sqrt{\sum_{f=1}^{F}\left(ASE_{dB}(l,f)\right)^2}} \qquad 1\le l\le L \tag{42}$$

where

$X(l,f)$ is the l^{th} normalise row of NASE matrix.

F is the number of ASE spectral coefficients.

L is the total number of frames.

4. The basis functions are calculated by applying a basis decomposition algorithm (for example PCA, SVD) on NASE matrix.
5. Calculating the ASP feature Y by multiplying NASE matrix with the basis function extracted in step 4, as equation 43 shows:

$$Y=\begin{cases} XV_E & \text{for SVD} \\ XC_E & \text{for PCA} \end{cases} \tag{43}$$

It worth mentioning that many of the short time Fourier transform frequency information were discarded during the calculation of NASE bins due to their lower frequency resolution. Figure 15 shows the ASP of a speech signal, the PCA is used to project the 32 dimension NASE matrix to 18 dimension ASP feature.

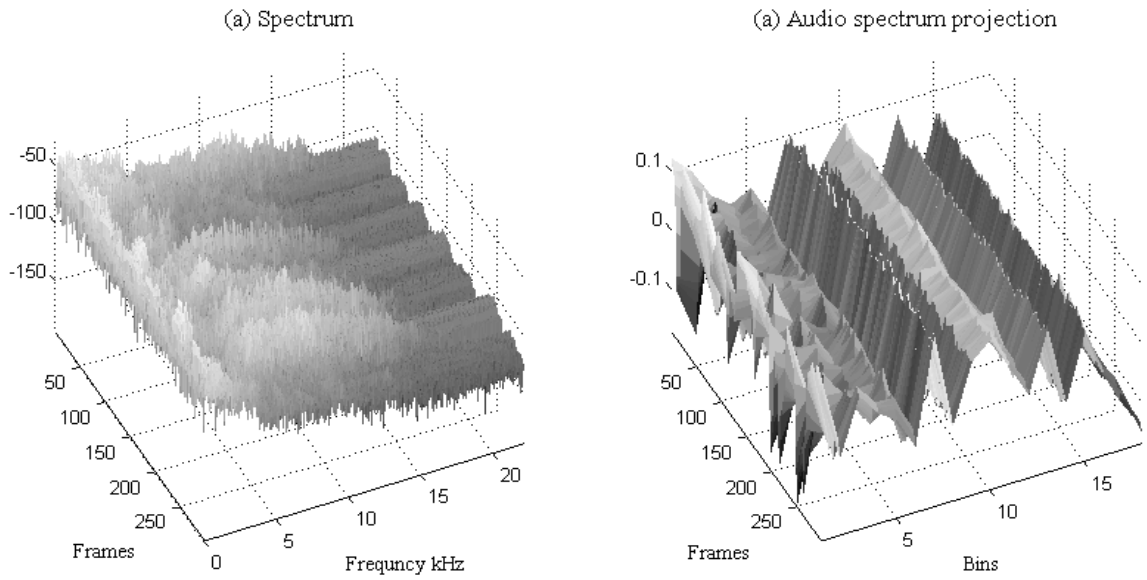


Figure 15: ASP of a speech signal (a) Signal spectrum (b) ASP feature of size 18.

Chapter 4: Machine Learning for Classification

4.1 Introduction

After discussing audio features in Chapter 3, the task in this chapter is to review the machine learning techniques that are utilised for audio classification during the development and the testing of the proposed technique.

4.2 Supervised Machine Learning

Supervised machine learning is the most appropriate technique to classify audio content into the three pre-defined classes of speech, music and environmental sound. The training process will guide the machine learning system to map the input features to the target class. Thus, with a good feature set, the machine learning technique becomes able to successfully identify the class of the sample being tested using the same feature set used for training.

There are many supervised machine learning techniques that can be utilised to perform audio content classification. The most common techniques in the field are the GMM, NNet and classification trees. The following sub-section will discuss each of these techniques.

4.3 Gaussian Mixture Model

A Gaussian mixture model (GMM) is a probability density function that is created by finding the weighted summation of multiple Gaussian components. A mixture of Gaussians is used to provide a class density that is richer than a single Gaussian. The summation of Gaussian densities is shown in equation 44 (Reynolds and Rose, 1995; Kim et al., 2005).

$$p(x \mid \lambda) = \sum_{m=1}^{M} c_m b_m(x) \qquad 44$$

where:

λ is the GMM parameters set.

M is the number of mixture components.

x is the D-dimensional data vector.

c_m is the weight of the m^{th} mixtures.

$b_m(x)$ is the m^{th} component Gaussian density.

Each component density is a D-variate Gaussian function that is represented by equation 45:

$$b_m(x) = \frac{1}{(2\pi)^{d/2}|\Sigma_m|^{1/2}} \exp\left\{-\frac{1}{2}\left[(x-\mu_m)^T \Sigma_m^{-1}(x-\mu_m)\right]\right\} \quad 45$$

where

μ_m is the mean vector.

Σ_m is the covariance matrix.

The mixture of these weighted Gaussians should satisfy the constraint in equation 46:

$$\sum_{m=1}^{M} c_m = 1 \quad 46$$

The covariance matrices, the mean vectors, and the mixture weights c_i for all component densities parameterise the final GMM. These parameters are represented by equation 47:

$$\lambda = \{c_m, \mu_m, \Sigma_m\} \qquad m = 1,...,M. \quad 47$$

The GMM is powerful because it has the ability to form smooth approximations to arbitrarily shaped densities. Figure 16 shows the way that GMM represents an audio features vector of a male speaker, in which ten Gaussian models were combined to create the mixture model that shapes the MFCC histogram (Reynolds and Rose, 1995; Kim et al., 2005).

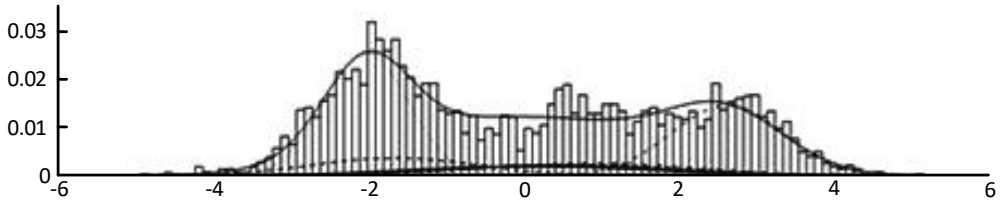


Figure 16: GMM representing an MFCC of a male speaker

To achieve a best-matching GMM (or maximum likelihood), when estimating λ to represent GMM parameters, the well-established maximum likelihood estimation method can be used.

For a sequence of *T* training feature vectors, $X = \{x_1, \ldots, x_n\}$, the GMM likelihood, assuming independence between the vectors, can be written as shown in equation 48:

$$p(X \mid \lambda) = \prod_{i=1}^{n} p(x_i \mid \lambda) \qquad 48$$

Because it is not possible to perform direct maximisation of the nonlinear function λ, the maximum likelihood can be calculated iteratively using a special case of the expectation-maximisation algorithm (Dempster et al., 1977).

The expectation-maximisation algorithm starts with an initial model λ and then estimates a newer model λ^{new}, such that $p(X \mid \overline{\lambda}) \geq p(X \mid \lambda)$. This process repeats iteratively until the convergence threshold is reached.

Equations 49, 50 and 51 are used to approximate the mixture weights, means, and variances respectively. These formulas allow a monotonic increase in the model's likelihood value.

$$\mu_m^{new} = \frac{\sum_{i=1}^{n} p(m \mid x_i, \lambda) x_i}{\sum_{t=1}^{n} p(m \mid x_i, \lambda)} \qquad 49$$

$$\sum_m^{new} = \frac{\sum_{t=1}^{n} p(m \mid x_i, \lambda)(x_i - \mu_m)^T (x_i - \mu_m)}{\sum_{t=1}^{n} p(m \mid x_i, \lambda)} \quad 50$$

$$c_m^{new} = \frac{1}{n} \sum_{i=1}^{n} p(m \mid x_i, \lambda) \quad 51$$

where value $p(m \mid x_i, \lambda)$ can be computed using equation 52 (Reynolds and Rose, 1995; Kim et al., 2005)

$$p(m \mid x_i, \lambda) = \frac{c_m b_m(x_i)}{\sum_{j=1}^{M} c_j g_j(x_i)} \quad 52$$

4.4 Neural Network

The neural network (NNet) is an artificial intelligence network that is inspired by the human brain. It contains a set of interconnected nodes known as neurons that work together to process the input values in order to produce the related output.

There are many types of NNet, such as Kohonen self-organising maps, the multi-layer perceptron, the time-delay NNet and the hidden control NNet (Kim et al., 2005). Among these types, the MLP is the most common NNet used for speech recognition and sound classification (Kim et al., 2005) and it will be adopted in this research.

MLP is a feed-forward network that has an input layer, zero, or any number of hidden layers, and one output layer. The number of neurons in the input layers is equal to the size of the training features vector while the number of output-layer neurons determines the number of output classes to be identified. The hidden layer could have any number of neurons.

Most of the MLP networks are fully connected, so each neuron is connected to all the neurons in the previous layer, and neurons are connected only if they lie in adjacent layers. This is shown in Figure 17:

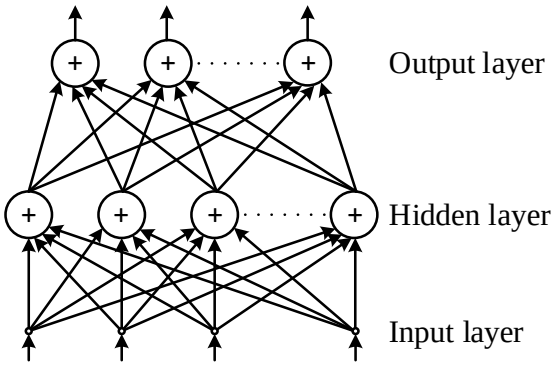


Figure 17: Multi-layer perceptron NNet

All connections except for output connections have their connection weight, and the neuron will be a function of these incoming weights. The simplest function is the weighted summation, but usually the neuron has a transfer function. The transfer function can be a linear discrimination function; such a function is suitable for linearly-separable classes (Ethem, 2014) as shown in equation 53:

$$s(x) = \begin{cases} 1 & if \quad a > 0 \\ 0 & if \quad a \le 0 \end{cases} \qquad 53$$

where x is the input vector.

For the nonlinear case, there are many transfer functions such as sigmoid, hyperbolic tangent, Gaussian and softmax functions. Among these, the sigmoid function illustrated in equation 54 is the most commonly used transfer functions (Kim et al., 2005; Brownlee, 2011):

$$s(x) = \frac{1}{1 + \exp(-x)} \qquad 54$$

where x is the input vector.

In training the multi-layer perceptron, a weights matrix is updated in order to achieve the desired output vectors from the corresponding input vector. The training is achieved usually via the error backpropagation algorithm which minimises the mean square error between actual output and the desired output. The use of mean square error in the back-propagation algorithm results in a training difficulties, including slow convergence and falling in local minima. Some other gradient-based optimisation methods have been proposed to improve training performance, such as the conjugate gradient, quasi-Newton and Levenberg–Marquadt methods. The Levenberg–Marquadt method has been selected to be used in training due to evidence of improved performance over the other methods (Webb et al, 1988; Webb and Copsey, 2011).

The advantage of NNet is that it relies on the ability to provide a good classification performance. Moreover, NNet requires only a few parameters. The disadvantage lies in the slow training procedure and the need of a re-training when a new class is added (Kim et al., 2005). The results in Chapter 9 show that NNet can outperform GMM.

4.5 Decision Tree Family

A decision tree (DT) is a tree-like structure that has nodes and edges; the edges represent attributes, the non-terminal nodes represent attribute test, and the terminal nodes represent the class labels. Generally, decision trees are used either for classification or regression. In this research, a tree-based approach is utilised as a feature selection technique for speech, music, environmental sound audio content classification. A decision tree is a simple model that adopts a multi-stage decision process. It uses a subset of the training features at each level of the tree to make the decision as shown in Figure 18:

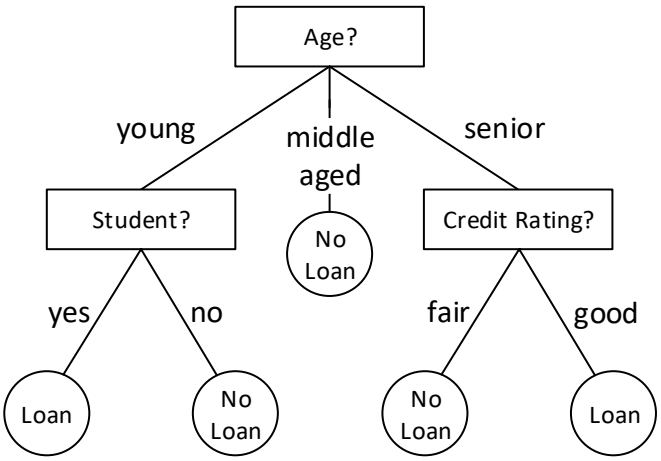


Figure 18: A decision tree for a loan example.

The tree has nodes representing features, edges representing features values, and leaves that represent class labels. The tree has a single root node that has no incoming edges and two or more outgoing edges, a number of internal nodes that have one incoming edge and two or more outgoing edges, and finally a leaf node that has one incoming edge and no outgoing edges. If all the nodes except the leaf nodes in the tree have two outgoing edges, then the tree is called a binary tree as Figure 20 shows.

From this point onward, only the binary tree will be discussed because any non-binary node can be implemented by using multiple levels of binary nodes. Furthermore, it is more straightforward for each node to choose a single threshold value that is used to split into two nodes (Breiman et al., 1984; Webb and Copsey, 2011).

Decision trees can represent complex linear or nonlinear decision boundaries via recursive partitioning as shown in Figure 19 that represent a linearly separable data by a line that is not parallel to the coordinates, despite this, the tree achieved a 100% classification accuracy on the dataset of both classes.

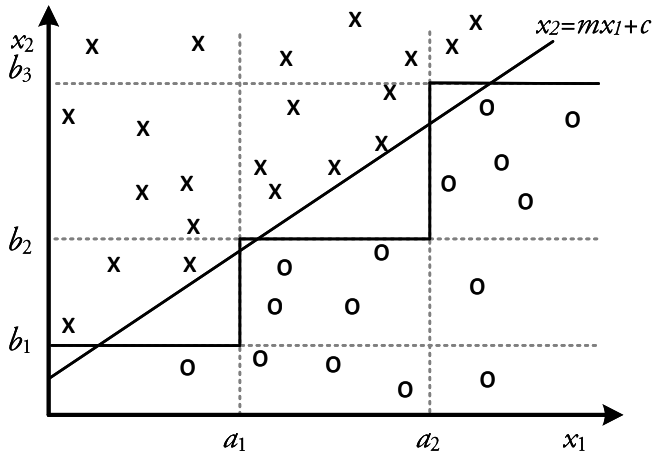


Figure 19: A two-dimensional data classification problem

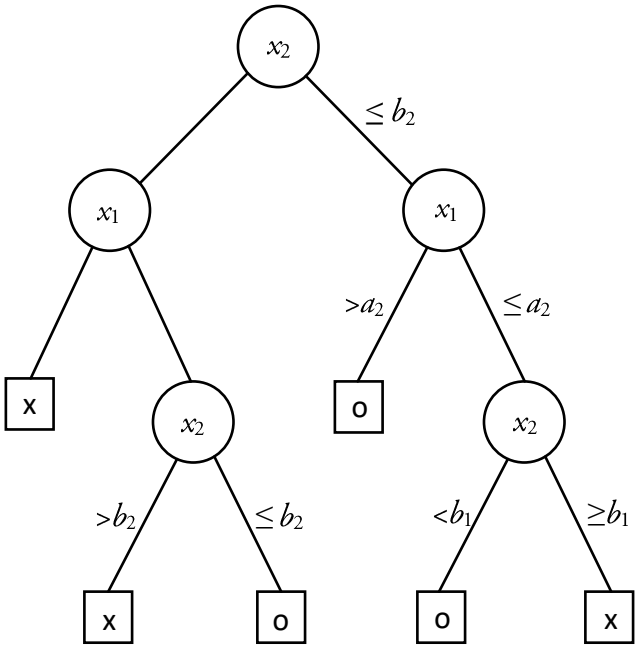


Figure 20: A binary decision for the data in Figure 19

4.5.1 Types of Decision Tree

Decision trees can be categorised into two types depending on the intended task. The first is the classification tree and the second is the regression tree. The major difference between these two types of construction lies in the way of measuring split performance. In classification, a good split should lead to a purer node, as discussed in the next paragraph. While in regression tree analysis, the impurity is measured by the mean square error from the fitting function (Ethem,

2014). The rest of this chapter will focus on classification trees only because it is the type of decision tree that is related to this work.

4.5.2 The Construction of a Classification Tree

A tree is constructed by determining the binary split of the data that will produce a child node that has a subset which is purer than the parent subset (Breiman et al., 1984; Webb and Copsey, 2011). Tree construction starts from the root node by successively partitioning the feature space. A labelled dataset will be used, as equation 55 shows:

$$L = \left\{ \left(\boldsymbol{x}_i, y_i, i = 1, \ldots, n \right) \right\} \qquad 55$$

where:

$\boldsymbol{x}_i$ is the i^{th} data sample.

y_i is the i^{th} class label.

n is the number of samples.

Tree construction involves the following three steps (Breiman et al., 1984; Webb and Copsey 2011):

1. Selecting a splitting rule for each internal node in the tree; this splitting is achieved by selecting a feature and a splitting rule that produces subset that is purer than the set in the parent node.
2. Determining the leaf node at which to stop splitting; each node needs to be checked to be a leaf node or to continue splitting. If splitting continues until each terminal node has pure class, this will likely produce a large tree that overfits the data. A better alternative is to construct a smaller tree that underfits the data; such a tree will give better results on the unseen data compared to the overfitted tree.

Several stopping techniques have been proposed in the literature: one technique is to stop splitting after reaching a specific number of levels; another technique is to determine a stopping condition; or by using an alternative technique suggested by Breiman et al. (1984) that works by growing the tree successively and then pruning the tree using cross-validation to choose a sub-tree with a lower misclassification rate.

3. Assigning class labels to each terminal node that minimises the misclassification rate.

These three steps will be discussed briefly in the following sections.

4.5.3 The Selection of a Splitting Rule

A splitting rule is the criterion at each node to divide the data into two sub-groups in order to achieve the desired classification pattern. The decision trees can represent complex nonlinear decision boundaries via recursive splitting.

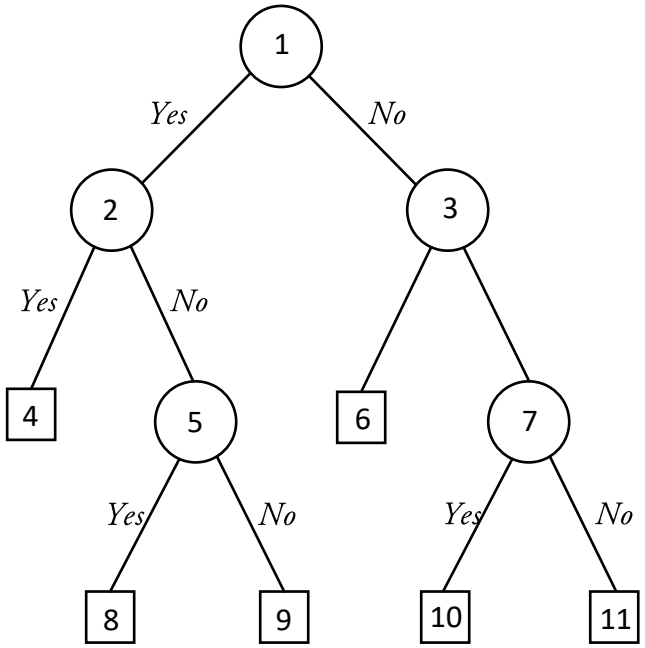
Figure 21 demonstrates the decision tree and decision regions of the binary tree presented in Figure 20.

The classification decision at each node depends on a single feature value, or a set of features for each node. The split rule is highly dependent on the nature of the features.

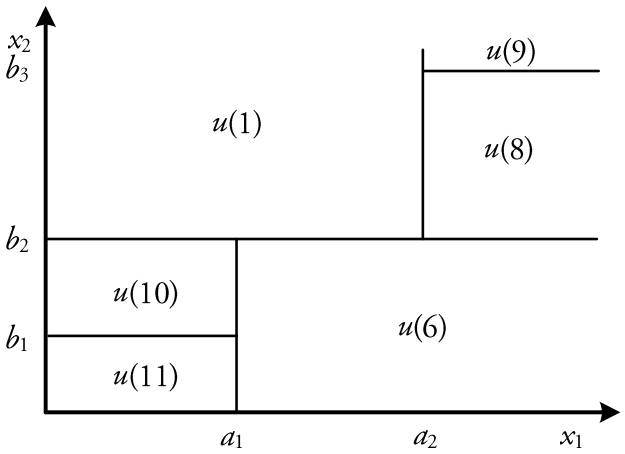
- Binary Feature: This type of feature will have one of two values, for example it can take either zero or one. The split in this case is the easiest: If it is the first value, then the left child node is selected, otherwise the right child node is selected.
- Nominal Feature: This type of feature can take many distinct values. In this case, the parent node can have a number of child nodes equal to the distinct values of the feature. These distinct values can be grouped to reduce the number of child nodes.
- Ordinal Feature: The case here is similar to the nominal feature, but the grouping should maintain the feature values order.

- Continuous feature: A split that consists of a condition on a single feature or a combination of features; a condition therefore is chosen to give binary or multiway split.

The scope of this research will be limited to continuous split rules only because all features mentioned in Chapter 3 are continuous.



(a)



(b)

Figure 21: (a) The binary decision for the data points in Figure 20, (b) the resultant decision regions

4.5.4 Multiway Split or a Binary Split

The multiway split at each node is suitable in specific cases, but most often it leads a rapid data fragmentation. As such, it is better to use a binary split, especially considering that a multiway split can be achieved by a sequence of binary splits. Therefore, only the binary split will be considered from this point onward.

4.5.5 Selecting a Split Condition

The task here is to split the input data at each node into purer subsets, which can be done by estimating node impurity. The impurity measures the extent of purity for a region containing data points from possibly different classes. Thus, the node impurity will have the maximum value when all the classes at node are equally likely, and a zero node impurity indicated that all patterns at the node are of the same class.

The probability that the class of a pattern x is ω_j, given that it falls into node *t*, is given by equation 56:

$$p\left(\omega_j \left| \boldsymbol{x} \in u(t)\right.\right) = p\left(\omega_j \left| t\right.\right) \sim \frac{N_j(t)}{N(t)} \qquad 56$$

where:

ω_j is the j^{th} class.

$\boldsymbol{x}$ is the feature pattern.

$u(t)$ is the t^{th} node in the tree.

$N(t)$ is the number of samples in the node $u(t)$.

$N_j(t)$ is the number of samples in the node $u(t)$ that belongs to the j^{th} class.

Equation 56 shows that tree node impurity for the class ω_j is proportional to the ratio of the total number of samples $N(t)$ to the number of samples $N_j(t)$ that belongs to ω_j. Breiman et al (1996) define the node impurity using equation 57:

$$I(t) = \Phi\left(p(\omega_1 \mid t), ..., p(\omega_C \mid t)\right) \quad 57$$

where Φ is a function defined on all C-tuples $(q_1, ..., q_C)$, that satisfy the condition $q_j \geq 0$ and has the following properties:

1. Φ is a maximum only when $q_j = 1/C$ for all j.
2. It is a minimum when for some j, $q_j = 1$ and $q_i = 0$ for all $i \neq j$.
3. It is a symmetric function of $q_1, ..., q_C$.

There are several used functions, which satisfy these conditions, for example (Hastie et al., 2016):

Gini measure:

$$I(t) = \sum_{i \neq j} p(\omega_i \mid t)\, p(\omega_j \mid t) = 1 - \sum_{i=1} \left[p(\omega_i \mid t)\right]^2 \quad 58$$

Entropy:

$$I(t) = -\sum_{i=1} p(\omega_i \mid t) \log_2 \left(p(\omega_i \mid t)\right) \quad 59$$

Classification Error

$$I(t) = 1 - \max_i \left[p(\omega_i \mid t)\right] \quad 60$$

The functions presented in equations 58, 59 and 60 can be used to find the impurity for a particular split. Figure 22 shows the plot for impurity function of a binary classification problem (C=2). The plot shows that the function has its minimum at $p = 0$ or $p = 1$ when all the patterns

belong to a single class, and has a peak at = $^1/_2$ when half of the samples belong to the first class and the other half belongs to the second class (Webb and Copsey, 2011).

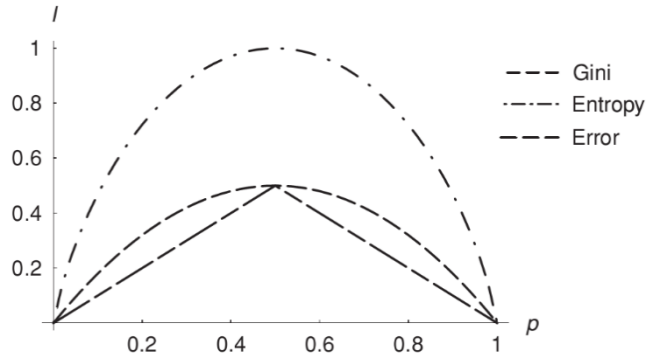


Figure 22: Node impurity function for binary classification.

Now the split performance can be measured by calculating the information gain ΔI by finding the purity difference between the parent node and all the child nodes in the tree, as equation 61 shows:

$$\Delta I(s_p,t) \triangleq I(t) - \sum_{i=1}^{k} I(t_i)\frac{N(t_i)}{N(t)} \qquad 61$$

where:

k is the number of child nodes.

$N(t)$ is the number of samples in the node $u(t)$.

$N_j(t)$ is the number of samples in the node $u(t)$ that belong to the j^{th} class.

I is the impurity measure function.

During the training phase, the best split gives the maximum ΔI, thus minimising the impurity of the child nodes. For the binary variable, the task is easy because there is only a single way to split. The nominal or ordinal variables many thresholds can be checked to find the optimum split that will maximise the information gain. The number of tested thresholds should, however, be kept reasonably small to prevent excessive computation.

There are some other splitting approaches reviewed by Safavian and Landgrebe (1991), however, the overall tree misclassification rate is not sensitive to the choice of a splitting rule if reasonable rules are employed (Breiman et al., 1984).

4.5.6 Terminating the Splitting Procedure

Theoretically, the tree may continue to grow by successively splitting nodes until each terminal node contains a single observation only, but this will end up with a very large tree that overfits the data. The overfitting will achieve an excellent classification accuracy on the training set, however, the result is expected to be poor on the testing set.

There are two approaches to overcome the overfitting problem in tree growing:

1. Setting an impurity threshold parameter: Splitting will stop if the node has an impurity lower than the threshold parameter. The problem here is how to set the threshold. Further, if the current node impurity is smaller than the threshold, then the next split might result in a lower impurity.
2. Pruning: The tree grows until the terminal nodes have pure (or an almost pure) class membership. After that, tree pruning starts by replacing a sub-tree with a terminal node with a class label equal to the class determined from the pruned sub-tree. This leads to a simpler and smaller tree, without decreasing the classification performance because all the pruned subtrees are not offering an improvement to the classification performance.

4.5.7 Assigning Class Labels to Terminal Nodes

After deciding that the node is a terminal node, a class should be assigned to it. This is an easy part of the process, in which the class that gives the smallest resultant misclassification is assigned to the node.

Thus, the task is to assign the terminal node to the class ω_k that minimises the misclassification m which is performed using equation 62:

$$m = \sum_{x_j \in u(t)} \lambda\left(y_j, \omega_k\right), \qquad \lambda\left(\omega_j, \omega_i\right) = \begin{cases} 1 & i \neq j \\ 0 & i = j \end{cases} \qquad 62$$

where:

x_j is the classification training pattern j

$u(t)$ is the part of the data space corresponding to the terminal node t.

$\lambda\left(\omega_j, \omega_i\right)$ is the cost function of assigning the pattern x to the class label ω_k, where x belongs to the class y_j, and both $\omega_k, y_j \in \{\omega_1, \ldots \omega_c\}$.

4.6 Ensemble Decision Classifiers

The term "ensemble classifiers", or "classifier fusion", refers to the combination of the results of multiple classifiers in order to improve classification performance. Ensemble learning achieved by (Hastie et al., 2016)

1. The training data is used to develop a set of base learners.
2. Combine the results of the trained learners to form the composite predictor.

Assume that there is a two-dimensional training set that contains two classes, and that the boundaries between these two classes data points are complex in a way that each classifier has some misclassified data points. Figure 23 shows data points that belong to two distinct classes C_1 and C_2. These data points are classified with two linear classifiers LC_1 and LC_2. Each classifier classifies the points below it as class 1, and the points above it as class 2.

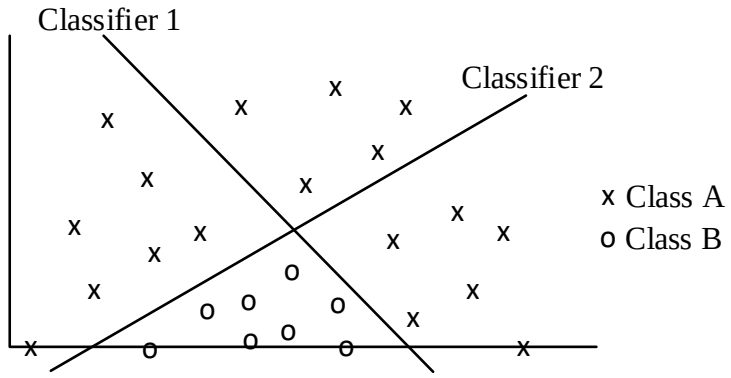


Figure 23: A binary classification problem two linear classifiers

It is clear that each classifier misclassifies some of the samples, but combining the result of these two classifiers with a simple rule as stated as pseudo code below:

```
IF LC1 AND LC2 predict the sample as Class B THEN
     Sample belongs to Class B
ELSE
     Sample belongs to Class A
END
```

This combination achieves 100% accuracy. The technique of combining multiple classifier results is not new. For instance, Kittler and Devijver (1982) combined the result of two classifiers, in which each uses different feature space. Chen et al, (1997) introduced the term "dynamic classifier selection"; and Hand et al (2001) introduced the term "classifier choice system". Both terms predict the optimum classifier for the given classification sample. On the other hand, the terms "classifier fusion" and "multiple classifier system" refer to the combining of the classification result of multiple classifiers to reach a single classification result (Gonzalez and E., 2008).

For a complicated classification problem with a high-dimensional feature space, the ensemble methods provide an efficient technique to overcome the training difficulties, because the overall classification result is achieved by combining the strengths of a collection of simpler base models (Hastie et al., 2016).

The two main categories of tree ensemble method are bagging and the random forest. Each of these categories has more than one approach. The sections below discuss the approach that has been followed to build each category.

4.6.1 Bagging

Bagging, which is sometimes called bootstrap aggregating (Breiman, 1996), is a procedure achieved by combining the results of multiple classifiers. Each of these classifiers are trained using a subset of training samples that are referred to as bootstrap samples. Bootstrap samples are produced by sampling an n replicates of the training set with replacements. Therefore, some of the patterns might be used several times (in the bag samples) and some other samples might not be used at all (out of the bag samples).

As a result, the new training set will have B bootstrap datasets of size n, each dataset is used to train a classifier. The final classification result is result that is most represented by B classifiers. Algorithm 1 below summarises the bagging steps (Webb and Copsey, 2011):

Algorithm 1: The bagging algorithm

1. Import the training set (x_i , z_i) where x_i is the i^{th} pattern, z_i is the label of the i^{th} pattern and i=1 to n.
2. Set B value to the number of bootstrap samples
3. For b=1 to B
   (a) Generate the boot strap sample of size n from the training set by sampling with replacement. Some patterns may be replicated and others may be omitted.
   (b) Train the classifier $\eta_b(x)$ using the b^{th} bootstrap sample as training data.
4. Classify the test pattern x using B number of trained classifiers, assign the class of the pattern x to the most represented classification result.

Bagging performance depends on the stability of the base classifier. Bagging becomes particularly useful by reducing the classification variance of an unstable classifier. If the classifier is stable, bagging will offer only a little improvement (Breiman, 2001; Skurichina, 2001)

The classifier is considered stable if the classification result is robust to minor perturbations in the training set. A classifier is unstable when a large change in classification result is caused by a small change in the training data. A non-stable biased classification module suffers from under-fitting and it tends to miss the relevant relations between the training features and target outputs. On the other hand, a high variance classification module suffers from over-fitting caused by modelling the noise within the training data, therefore becoming sensitive to small variations in the training set. The concept of bias-variance can be visualised graphically in Figure 24. An example of unstable classifier is NNet and classification trees (Hastie et al, 2001; Webb et al, 1988).

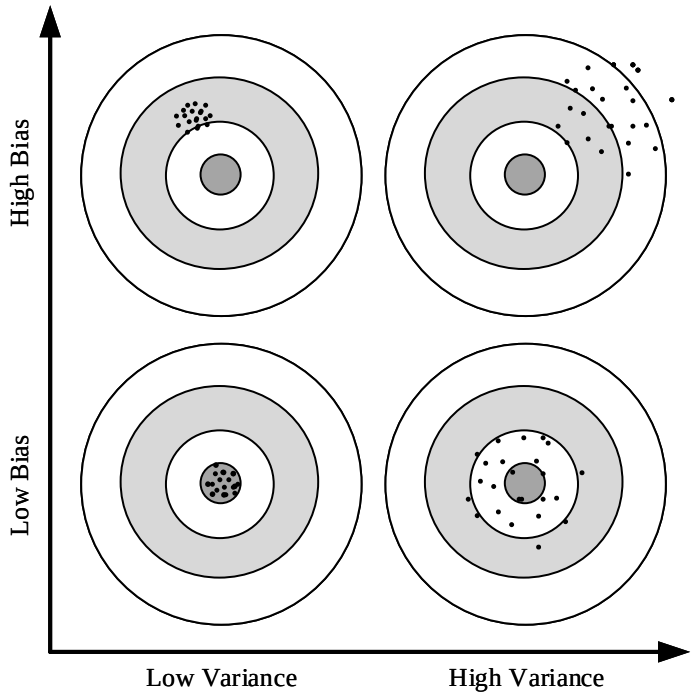


Figure 24: Graphical visualization of bias and variance

A vital aspect of the bagging procedure is the production of unstable classifiers. For a given bootstrap sample, the probability of including a pattern in the training set is equal to $1-(1-1/n)^n$. For large values of n, the probability will be approximately equal to $1-1/e = 63.2\%$. Therefore, each bootstrap sample will contain only about 63% of unique samples. As a result, a set of different unstable classifiers will be built.

Depending on the stability of the classifier, bagging performance differs. In the case of an unstable classifier like NNet and classification trees, bagging reduces the variance and improves the classification result. In the case of a stable classifier (robust to minor changes in the training set), bagging may not be able to improve the performance (Webb and Copsey, 2011).

4.6.2 Random Forest

Random Forests (RFs) is a modified version of bagging (Breiman, 2001). Both RFs and bagging use bootstrap training samples to train multiple classifiers. During classification, the result of all trained classifiers are combined using a majority-vote decision rule.

The fundamental difference between bagging and RFs lies in the training features. RFs improves classification variance by reducing correlation between trees. This is achieved by selecting different features in subset *m.* These features are selected randomly for each tree node in the forests where $m<M$ and M is the total number of available classification features (Breiman, 2001; Webb and Copsey, 2011).

To train the RFs, a pre-defined number of trees are created and each tree in the forest is grown by determining the best split at each tree node using one of the *m* features to construct the tree. All the trees in the forests are trained using a different bootstrap sample, and each tree is fully grown without pruning. For classification, the majority of tree class voting is assigned

to the classified pattern. The RFs algorithm is summarised in Algorithm 2 (Webb and Copsey, 2011; Hastie et al., 2016).

Algorithm 2: The random forests algorithm.

1. Import the training set (x_i , z_i) where x_i is the i^{th} pattern, z_i is the label of the i^{th} pattern, and i=1 to n.
2. Set B value to the number of trees in the forest and m to the number of features to be used in each tree, m should be smaller than the total features number.
3. For b=1 to B
   (a) Generate the boot strap sample of size $n \times m$ from the training set by sampling with replacement. This results in omitting some of the features in each tree. Also, some patterns may be replicated and others may be omitted.
   (b) Train decision tree classifier $\eta_b(x)$ using the b^{th} bootstrap sample as training data, consider the best split among the randomly selected m features at each tree node.
4. Classify the test pattern x using each of the $\eta b(x)$ trained classifiers, and assign the class of the pattern x to the most represented classification result.

Each tree in the forest will ignore some training samples as a result of bagging and some of the training features as a result of tree subset feature selection. Because of this, the RFs is simple to train and easy to tune. The random forest also shows the ability to handle a very large number of features. Testing shows that RFs classification accuracy is comparable to the best classifiers currently available on many datasets. As a result of these features, RFs is a popular method and has been adopted in a variety of packages (Breiman, 2001; Webb and Copsey, 2011).

Breiman et al. (2001) state that RFs cannot overfit training data. Segal (2004) demonstrates small gains in performance can be achieved by controlling the depths of the trees that are grown in the random forest. But Hastie et al. (2016) demonstrate that utilising full-grown trees will rarely add cost to any fitting problem and suggested the use of fully-grown

tress is a better option than having more tuning parameters of tree depth, according to that the trees will be fully grown during testing and evaluation.

The source of the randomness in the RFs is achieved on node level during training, so while the tree is grown the feature that is used to split the data between the two child nodes is selected randomly from the available boot strap features. The split is performed to achieve the maximum possible purity in the child nodes, due large numbers of utilised trees in the forest and the full tree expansion (until each node have a pure or semi pure class content), each feature will be utilised many times in the forest. The way that each feature affect node purity can give an efficient indicator to features importance. This is also the reason that the tree is not affected by the irrelevant features, so if an irrelevant feature selected to split the node content and the best possible split could not improve the child nodes purity, all what is required is to add another level to these nodes and the randomness will allow the utilization of other features that can perform better until the terminal nodes in this sub tree have a pure or semi pure class content.

All these characteristics of the RFs make it an excellent choice for high-level audio content classification and feature importance ranking that can be adopted to find the reduced set of optimum features.

Chapter 5: Feature Dimensionality Reduction

5.1 Introduction

This chapter discusses two approaches of feature dimensionality reduction; the first is the feature selection and the second is the feature extraction, showing advantages and the major differences between them.

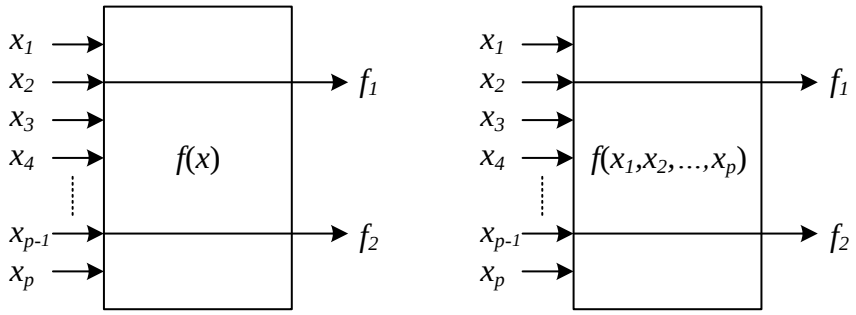
5.2 Feature Selection and Features Dimensionality Reduction

Both of feature selection and feature extraction aim to reduce the dimensions of feature space by removing potentially redundant or irrelevant data. This, in turn, leads to:

1. Improved classification performance.
2. Improved learning efficiency.
3. Removal of irrelevant data.
4. Reduced classifier complexity.
5. Reduced computational and storage overheads in both training and classification phases.
6. Improved classification performance, especially for real-time classification systems or for applications that use limited hardware resources.
7. Reduced cost of future extraction.

Therefore, using a small, yet efficient feature set is one of the most important aspects of classification. However, feature importance is highly dependent on the data to be processed. This includes the content type, the target classification task -as different classes have different optimal feature sets, the source and sample quality. All these variables make the manual feature selection that is based on user experience and the psychoacoustics properties inefficient method due to the huge amount of combinations that need to be tested.

The main difference between feature selection and features extraction is that in feature selection, the variables that do not contribute to class separability will be ignored as shown in Figure 25a in which only a subset of *p* input features is selected, while in feature dimensionality reduction, the *p* input features are transformed to a lower dimensional feature space as Figure 25b shows. In this way, a new set of features will be created from the input features. Therefore, it is considered as feature extraction method. The transformation introduced by feature extraction can be linear or nonlinear, supervised or unsupervised (Webb and Copsey, 2011).



(a) Feature selection (b) Feature extraction

Figure 25: Dimensionality reduction using (a) feature selection, (b) feature extraction.

To summarize, both feature selection and feature extraction can be considered as transformations that apply a set of weights to the input features in order to obtain the reduced transformed variables. However, the difference is that the weights in feature selection are binary -either 0 or 1-, so that original features are retained, while the weights of feature extraction are continuous, so the resulted output represents a new feature set extracted from the input features (Molina et al., 2002).

5.3 Features Relevance and Redundancy

The features used in a particular classification task can be categorised according to their irrelevance and redundancy into the following four categories (Webb and Copsey, 2011; Tang et al., 2014):

1. Strongly relevant features: These features are highly related to the classification task. Thus, it has a direct effect on classification performance. If these features are removed, the classifier performance will drop dramatically.
2. Weakly relevant features: These features are related to a subset of the classification samples only. If these features are removed, classification performance will be low only for a subset of the samples.
3. Irrelevant features: These features are not related to the classification task. The removal of these features will not affect classification performance at all.
4. Redundant features: These features are related to the classification task but they are redundant. Thus, removing them does not affect classification performance.

A good classification features set should include all the strong relevant features, a subset of the weakly relevant features, and none of the irrelevant or the redundant features.

5.4 Feature selection

Feature selection methods can be categorised into the following three categories:

1. Filter methods: Filter methods filter out the poorly informative features depending on their statistical properties such as distance and dependency. Filtering is achieved without utilising a classification algorithm.
2. Wrapper methods: The wrapper methods are classifier dependent; multiple subsets of the features are evaluated using the classifier. The evaluation of a new subset will stop when the desired performance is reached. Because of this, they are more computationally demanding than filter methods. However, they perform better for feature selection.
3. Embedded methods: Embedded methods of feature selection are built in classifier design. Thus, it is classifier dependent. An example of the embedded method is a decision tree

that in many cases when it is trained it will utilise a subset of the features and this is considered as a built-in feature selection algorithm.

So the filter methods do not incorporate any learning. The wrapper methods use the learned classifier to measure the quality of subsets features selection without the need of incorporating a knowledge about the classifier structure, therefore it can be combined with any machine learning classifier. While in the embedded methods the learning and the feature selection are related and done together as a single part and cannot be separated.

Each one of these three methods has its advantages and disadvantages. For example, filter methods are computationally efficient but usually have poor performance when compared with wrapper methods (Kohavi and John, 1997; Liu and Motoda, 1998). The exhaustive search of the wrapper methods imposes a huge computational overhead, especially with a large feature set. Different search strategies (Liu and Motoda, 1998) and also genetic algorithms (Guyon and Elisseeff, 2003) were utilised to improve the wrapper methods performance.

The wrapper method is utilised to evaluate the implemented features selection technique, it has been chosen in order to give the ability to create a feature set that can be generalized for the studied classes allowing other researchers in the field to adopt presented feature list and ranking to utilise them with their choice of machine learning technique.

To evaluate feature selection in order to find the best possible feature subset there are serval measurement that can be utilised, an example of these measurements are:

- Measures rely on data general properties, for example the feature ranking (ranked by a metric if the archived result score below a predefined threshold the feature is eliminated). Interclass distance (measure the distance between each class member) and probabilistic distance (depend on the probabilistic distance between the classes).

- Measures that rely on classifier are designed in which the features that improve the classifier performance is used. These measures can be used with the wrapper or the embedded method only.
- Measures rely on classification accuracy of the classifier that is trained using trained using feature subset.

The classification accuracy is the adopted measure because it directly measures the classification accuracy of the selected feature subset. To perform feature ranking, the node information gain in combination with RFs is utilised as classifier design measure to rank according to their importance as mentioned in Chapter 4.

Finally, the sequential backward selection (or elimination) is utilised to find the best feature subset for classification. In the sequential backward selection, start with the full feature set and each time the lowest ranked feature is deleted until reaching the smallest optimal or near optimal features set.

5.5 MPEG-7 Feature Dimensionality Reduction Techniques

MPEG 7 has not proposed feature selection techniques, but it utilises feature dimensionality reduction techniques that include the PCA and SVD. Before discussing each of these techniques, the calculation of Eigenvectors and Eigenvalues are illustrated first.

5.6 Eigenvectors and Eigenvalues

Assume that the matrix $M_{i\times d}$ represents i number of data points in the d-dimensional coordinate system. There is a matrix $T_{d\times n}$ that can project the data points into a new n-dimensional orthogonal coordinate system. The new coordinate system has the greatest

variance between the data points on the first coordinate, the second-greatest variance on the second coordinate, and so on as equation 63 illustrates below. Each column of this transformation matrix can be referred to as an Eigenvector.

$$M_{i\times d}T_{d\times n} = M'_{i\times n} \qquad 63$$

where:

d is the number of dimensions of the original data points.

n is the number of principal components dimensions of the resulting dataset.

$M_{i\times d}$ is i number of data points in the d-dimensional coordinate system.

$T_{d\times n}$ is the transformation matrix, that has d number of rows and n number of column "Eigenvectors".

$M'_{i\times n}$ is i number of data points in the orthogonal n-dimensional coordinate system.

These Eigenvectors can be found for square matrices, although some square matrices have no Eigenvectors. The Eigenvectors have the following properties (Kim et al., 2005; Chapra and Canale, 2006; Jordan et al., 2000):

1. Eigenvectors are orthogonal and they are usually sorted according to their variance.
2. The length of a vector does not affect whether the vector is an Eigenvector or not.
3. The Eigenvectors are usually scaled to have a length equal to one in order to avoid the scaling after data decorrelation.
4. Each Eigenvector that is calculated for a square matrix A has its own associated Eigenvalue for the corresponding matrix A, as shown in equation 64:

$$Ax = \lambda x \qquad 64$$

where:

A is a square matrix.

x is an Eigenvector corresponding to λ.

λ is a scalar that represents the Eigenvalue of A.

The higher Eigenvalue represents the higher variance in the data point, thus the Eigenvector with the highest Eigenvalue represents the principal component.

Eigenvectors and Eigenvalues Calculation

There are two methods to calculate the Eigenvector and the Eigenvalue; the first is using the analytical method, while the second is by using an unsupervised machine learning method. Each method has some advantages and some disadvantages. The unsupervised machine learning method utilises a single layer NNet. Convergence is very fast, but there is a chance of having training difficulties with large datasets (Haykin, 1998). On the other hand, the analytical method calculates the exact values directly but it requires a higher computational power. This can be overcome by using numerical methods. In this research, the analytical approach is adopted to prove the concept without having the risk of incorrect calculation due to convergence difficulties.

The first step is to find the Eigenvector and the Eigenvalue. The analytical method is to find the covariance between all the dimensions of the data points using equation 65 to find how much these two dimensions vary from the mean with respect to each other.

$$\operatorname{cov}(a,b) = \frac{1}{n-1}\sum_{i=1}^{n}\left(d_a(i)-\bar{d}_a\right)\left(d_b(i)-\bar{d}_b\right) \qquad 65$$

where

n is the input data point dimensions.

d_a, d_b are two of the input data point dimensions.

For the data points that have more than two dimensions, it is possible to calculate a covariance matrix by measuring the covariance between any two given dimensions using equation 65, and presenting the result as a single two dimensional square covariance matrix of size equal to the dimensions of the input data points. For example, for *n*-dimensional data points, the covariance matrix can be defined by equation 66.

$$C = \begin{pmatrix} cov(d_1,d_2) & cov(d_1,d_2) & \cdots & cov(d_1,d_n) \\ cov(d_2,d_1) & cov(d_2,d_2) & & cov(d_2,d_n) \\ \vdots & & \ddots & \\ cov(d_n,d_1) & cov(d_n,d_2) & & cov(d_n,d_n) \end{pmatrix} \quad 66$$

The variable d_i represents the i^{th} dimension of the input data points. $cov(d_i,d_j)$ represent the covariance between the i^{th} and the j^{th} feature. Because the multiplication operation is commutative, the covariance becomes symmetrical so that only the upper triangle values can be calculated. These can then be copied to the lower triangle. The non-singular and symmetric covariance matrix leads to the orthogonality of the principal component (Chapra and Canale, 2006; Strang, 2009).

Now it is possible to start the calculation of the Eigenvalues and Eigenvectors for the covariance matrix *C*. The relationship can be expressed in the format of equation 67:

$$(C - \lambda I)X = 0 \quad 67$$

where

λ is the Eigenvalue vector.

I is the identity matrix.

X is the Eigenvector associated with an Eigenvalue.

The Eigenvalue can be found if C satisfies the characteristic equation 68.

$$\det\left(C-\lambda I\right)=0 \qquad 68$$

After finding the Eigenvalues, they can be substituted into equation 67 to find the Eigenvectors. The resultant Eigenvectors would probably not have a length equal to one (not a unit vector). This will cause a scaling to the data points during feature space transformation; to avoid this scaling in the data points, the resultant Eigenvectors can be scaled to the Eigenvalue's unit Eigenvector using equation 69:

$$X_i^u=\frac{X_i}{\sqrt{\sum_{i=1}^{n}X_i^2}} \qquad 69$$

where:

n is the vector length.

X_i^u is the resultant i^{th} element of the unit vector.

X_i is the i^{th} element of the input vector.

5.7 Principal Components Analysis

One method of data decorrelation is principal components analysis (PCA) which was introduced by Pearson (1901). The aim of PCA is to transform the data into a new orthogonal coordinate system that has the axes, or principal components, ordered according to the variance of the data. PCA can reduce the redundancy and the mutual information in the feature space. This allows dimensionality reduction while preserving most of the information at the same time identifying the feature importance. However, the problem with PCA dimensionality reduction is that PCA might miss out some important information due to low variance, and in some other cases it might not provide any improvement (Harrington, 2012).

To apply PCA transformation, the matrix T needs to be calculated in order to transfer the d-dimensional feature space to the orthogonal $n \leq m$ dimensional principal component as shown in equation 63. The proportion of the variance can be used as an indicator to select the subset of principal components.

The transformation matrix T is found by following these steps:

1. Subtract the mean from each data points' d dimensions to produce zero-mean data.
2. Calculate the covariance matrix using equation 65.
3. Calculate the Eigenvectors and Eigenvalues of the covariance matrix.
4. Sort the Eigenvectors according to their Eigenvalues using equations 67 and 68.
5. Scale the Eigenvectors so that each vector has a unit length using equation 69.
6. Select the top n principal components that have the highest Eigenvalues to form the transformation matrix $T_{d \times n}$.
7. Apply equation 63 by multiplying the data points by the resulting transformation matrix T, in order to transfer the data points to the new dimensions of the orthogonal principal components.

5.8 Singular Value Decomposition

The singular value decomposition (SVD) is a factorization of a real or complex matrix. This factorization enables the identification and ordering of the dimensions so that data points show the most variation. This will increase the possibility of finding the best approximation of the original data points using fewer dimensions.

For a real matrix A, it is always possible to decompose it into three matrices as equation 70 shows:

$$A = U \Sigma V^T \qquad 70$$

where

A is the features matrix of size $L \times F$

U is a left singular vector a matrix of size $L \times L$.

Σ is a diagonal value matrix of size of $L \times F$.

V is a right singular vectors matrix of size $F \times F$.

For both matrices U and V, the columns are both orthogonal and orthonormal, and the matrix $\sum$ is diagonal of positive elements that are sorted in a descending order.

The following steps are applied to calculate SVD:

1. Find A^T and $A^T A$.
2. Determine the Eigenvalues of $A^T A$, finding the square root of the absolute Eigenvalues values, and sorting them in the diagonal matrix $\sum$.
3. Use the Eigenvalues from step 2 to determine the Eigenvectors of $A^T A$. and construct the matrix V from these Eigenvectors
4. Transpose V to get V^T.
5. Find U using equation 71:

$$U = AV\Sigma^{-1} \quad 71$$

where $\sum^{-1}$ is the inverse of matrix $\sum$ calculated in step 2.

The dimensionality reduction is achieved by reducing the number of columns of the matrix V to create the matrix V_E of size $F \times E$ where $E<F$, as illustrated in equation 72.

$$A_{red} = USV_E \quad 72$$

where

A_{red} is the reduced feature matrix of size $L \times E$

V_E is the reduced matrix V of size $F \times E$

This way, the SVD transformation produces decorrelated and dimension-reduced bases for the data.

Chapter 6: Architecture of New High-Level Audio Content Classification & Feature Selection

6.1 Introduction

In the preceding sections, the theoretical background and the required components for classification and feature selection were discussed. Given the background covered, it is now possible to introduce the architecture of the implemented audio classification and feature selection technique which forms the subject of this thesis. This chapter reviews the layout of the implemented feature selection and classification technique.

6.2 High-level Audio Content Classification and Feature Selection System Architecture

The aim of this research is to perform the ranking and the selection of a subset of features that are the most suitable for the intended high-level audio content classification task. In order to evaluate feature selection performance, it is important to compare the classification performance before and after feature selection to evaluate how the feature selection affects classification performance.

The studied high-level classification covers the three classes, namely, speech, music, and environmental sound. Thus, the classification problem is a multi-class classification. The employed classification techniques allow multi-class classification. It is possible, therefore, to employ a single classifier to handle all three classes at once. Alternatively, it is possible to convert this classification problem into multiple binary classifiers and add one final stage of collective decision making. Such approach will simplify the intended classification task by training each module to detect a single class content of the three pre-defined classes.

Although the utilization of binary classifiers requires the training of C number of classifiers, where C is equal to the number of the target classes, each one of these C

classifiers is trained to discriminate a given class from the other $C-1$ classes. During testing, each test sample needs to be classified by each of the classifiers and collective decision making needs to be introduced to give the final classification result. This topic is discussed in detail in section 6.9.

The advantage of transforming the problem to a binary classification problem can be summarised in the following:

1. Each of C classifiers will handle simpler classification task, thus the training of these classifiers becomes faster and each one of them will consume fewer resources than a single multi-classifier; this is not a major issue especially in the case of utilisation of bagging and random forest that uses hundreds if not thousands of classifiers to perform the classification as mentioned in Chapter 4.
2. The processing power is consumed during the training phase unlike classification phase, which requires much less processing power. This will not be an issue in the studied case because training will be performed only one single time. After that, the trained modules will be used for classification that will not cause a heavy computational overhead.
3. The intended task is to classify a stream of audio samples into one of the C pre-defined classes. In such cases, a post processing stage is important even for a multi-class classifier in order to improve the classification result. This can be achieved by smoothing the classification pattern and making it similar to the real life audio content that has the same class content for many contagious frames. This topic is discussed in detail in section 6.7.
4. If a new class or classes are required to be added to a single multi-classification module, the training then needs to be started from the beginning. This might affect the whole classification performance for each individual class. On the other hand, if

binary classifiers were utilised, there is no need to re-train the available modules. The only required training is performed on the new class(es) module(s) that are required to be added. This will not affect the performance of the already established classifiers modules.

5. Similar to the previous case, it is easy to remove some of the available *C* classes in case of a binary classifier. Nevertheless, this is not the case if a single multi-classification module was utilised. This is a useful feature if the intended application requires a smaller number of classes, such as a speech only detector or a speech/music discriminator.

Therefore, the implemented system architecture utilises the binary-classifiers to perform the multi-classes classification. There are several approaches to decompose a classification problem of *C* classes into set of binary classification problems, for example:

1. One-versus-all: In this approach, the single multi-class classification problem is reduced into multiple binary classification problems. So, if *C* represents the number of classes in a multi-class classification problem, a *C* number of binary classifiers is required in which each classifier will discriminate a given class from the other *C*-1 other classes (Bishop, 2006; Theodoridis and Koutroumbas, 2009). During classification, the classifier that produces the maximum result is considered as a winner. The performance of this approach is comparable to more complicated approaches if the binary classifiers were well-tuned (Rifkin and Klautau, 2004).
2. All-versus-one: This approach is similar to the One-versus-all approach but it uses binary classifiers to distinguish between each two pair of classes. This requires the use of $C\left(C-1\right)/2$ binary classifiers. During classification, the class that receives the maximum votes is the winner (Bishop, 2006; Theodoridis and Koutroumbas, 2009).

3. Error correcting coding: This approach trains an N number of classifiers to detect the C classes. Initially, the matrix of size $C \times N$ is generated so that the C^{th} row of the matrix represents the C^{th} class codeword and the N^{th} column represents the target classification pattern of the N^{th} classifier matrix. During classification, the classification result is compared to each class codeword using a distance measure function and the winner class is the class that scores the shortest distance to the classification pattern (Bishop, 2006; Theodoridis and Koutroumbas, 2009).

The one-versus-all approach is the adopted approach in this research because it is the most suited for the intended classification task and it allows to utilise the post processing stage of collective decision making that improved the classification result significantly as the result in Chapter 7 shows.

To accomplish the intended feature selection and classification tasks, the system architecture illustrated in Figure 26 was developed. The main building blocks of this system are described briefly in this section.

1. Feature extraction: To extract the input audio file features. This requires pre-processing steps of normalisation and framing to prepare the signal for frame feature extraction. These tasks are discussed in this chapter.
2. Classes features ranking: Features ranking is performed for each class separately. To perform features ranking, the RFs is utilised to calculate the feature importance and rank the features according to their importance, which will be discussed in detail in Chapter 7.
3. Feature selection: By examining the class feature importance produced in step 2, a new set of reduced features set is selected. This will be discussed in detail in Chapter 7.

4. Class frame detector training: Train the classification module using the reduced features that were selected in step 3. The modules are trained to detect the existence of specific class content. This will be discussed in detail in Chapter 8.
5. Detection pattern smoothing: This stage is introduced to smooth the class detection patterns and forward it to the collective decision making stage.
6. Collective decision making: The collective decision making will combine the classifier detection patterns produced by the *C* classifiers in step 4 in order to produce a single stream of frames classification result by assigning one of the *C* classes label to each frame. This will be discussed in detail in Chapter 8.

The rest of this chapter provides a more detailed discussion on these six stages.

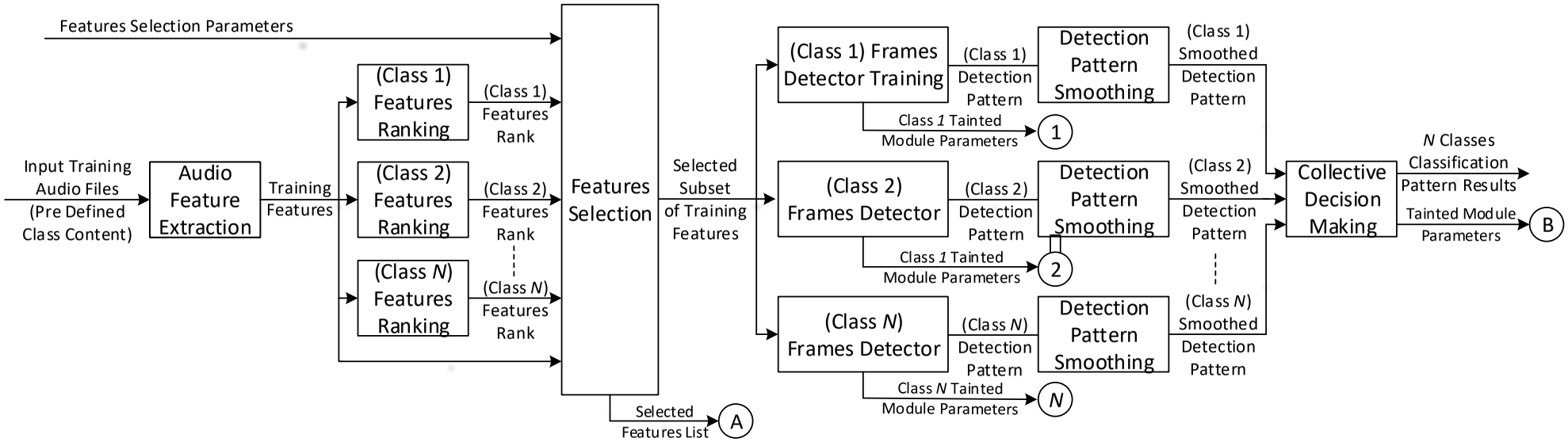


Figure 26: Block diagram showing the architecture developed for high-level audio classification and feature selection techniques

6.3 Audio Feature Extraction

Feature extraction is an essential stage to reduce the frame dimensionality. The dimensionality reduction is achieved by finding unique frame features that enable the classifier to identify the corresponding class of the frame. The low-level audio features are selected to comply with the MPEG-7 standard.

The advantage of utilising frame level audio classifier is the ability to process real-time data because it can perform classification in one single pass. In contrast, homogenous segment classification requires two phases of processing; the first phase is the homogenous segments boundary detection that is assumed to contain the same types of audio information followed by a second phase segments content classification.

The disadvantage of using a frame by frame classification is that a frame level classification might not produce a constant classification result for a homogenous segment. Thus, some frames can be misclassified as other class frames. This difficulty can be overcome by introducing a post-processing stage that improves the performance of the classifier by considering the classification pattern over a series of frames. This topic is discussed in detail in sections 6.7 and 6.8.

The utilised audio feature extraction contains the following three stages:

a) Normalisation, which normalises the input audio level, improving the performance of the next stage.
b) Framing, which splits the normalised audio file into frames.
c) Frame feature extraction, which extracts frame features from each non-silent frame.

Figure 27 shows the block diagram of the classification technique stages. A detailed description will be given by discussing the inputs, the processing and the outputs of the stage in the following sections.

Figure 27: Block diagram of the classification technique stages

6.3.1 Normalisation

Input file normalisation is an important pre-processing stage that aims to normalise the audio file to the peak level. This stage helps to extract features more consistently over different audio files by eliminating different recording conditions such as room acoustics, microphone type and placement. (Sezgin et al., 2011), leading to more constant frame features that allow more efficient training and more accurate classification result, equation 73 is used for normalisation:

$$s' = \left(\mathrm{s} - 2^{n-1}\right)\left(\frac{2^n - 1}{\max\left(\left|s - 2^{n-1}\right|\right)}\right) + 2^{n-1} \qquad 73$$

where:

n is the sample bit depth.

s is the unsigned wave audio samples.

s' is the normalised audio samples.

6.3.2 Framing

Overlapping frames are used, so the input audio file is split into overlapped frames of size 30ms with hop size of 10ms as stated in the MPEG-7 standard (Kim et al., 2005). The input is a single audio file and the output is *N* number of frames of size *M*.

6.3.3 Frame Feature Extraction

All the features discussed in Chapter 3 are extracted from the audible frames in the input audio files. Feature extraction will be achieved through the following two steps:

a) Audible frames detection: In this step, the frames are separated into two sets: the first set contains the audible frames; and the second set contains the silent frames. The AP value is used to determine whether the frame is audible or silent.

   Features are extracted from audible frames only. This simplifies the training process and improves the classification performance. If silent frames were included in the training data, then this will require the classification module to map identical silent frames to two distinct target classes since silent frames have the same features despite the content types of adjacent frames. This in turn leads to training difficulties. Therefore, the silent frames are excluded in order to reduce the training time and improve classification accuracy, as the test results show in Chapter 9.

b) Frame feature extraction: In this step, all the features discussed in Chapter 3 are extracted from audible frames of the input.

   Feature extraction is a time-consuming process in both training and classification phases. As such, feature selection has a huge effect on the computational power and speed of classification system.

   The utilisation of a small number of features might not allow the capture of the distinct class properties that will negatively affect the classification performance. This emphasizes the importance of feature selection. Efficient feature selection has a critical impact on the performance and efficacy of the classification system by utilising a small but highly related feature set in classification.

   Figure 28 shows the block diagram of the feature extraction steps.

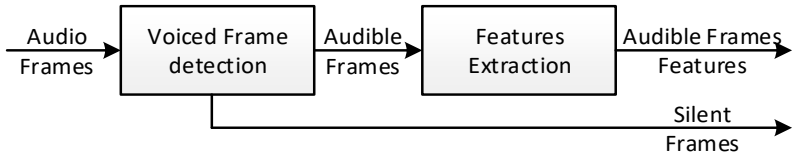


Figure 28: Block diagram of feature extraction step

6.4 Feature Ranking

The task here is to take frame features as an input and calculate their importance for the intended classification task, and produce a list of all features and their importance as an output for each individual class.

The RFs based feature ranking approach was used because it allows to track the classification performance by examining each feature to show how it affected the node purity and how it contributed to reach the final classification result of the RFs. The other reason behind selecting the RFs is that the RFs classification results outperformed the other tested classification modules in both classification accuracy and stability. Moreover, the RFs has been proven to have the capability not to overfit a small training set. So, feature ranking can be generalised for a larger dataset efficiently. This will be discussed in details in Chapters 8 and 9.

6.5 Feature Selection

The input for this stage is the feature selection parameter and the feature importance of each target class. The output is a subset of input features that can be used by class detectors in the next stage without compromising their classification performance. Feature selection determines the minimum possible set of features that do not compromise the required classification accuracy. Feature selection technique is discussed in detail in Chapter 8.

6.6 Class Frame Detector

This stage produces an individual class detection pattern that indicates whether the frame belongs to the specific class or not. The input for this stage is the subset of selected features. The outputs on the other hand, are the class detection pattern for each of the *N* classes that are fed to the smoothing stage, and the trained classification module that is used for classification after finalising the training and feature selection.

BT, RFs, GMM and NNet are selected to perform the classification; these machine learning techniques are the most used in the literature, and they are appropriate for the intended classification task. After training, the classification performance is evaluated by assessing the classification accuracy using the testing sample set.

The classification module is trained using a supervised algorithm; the training is done using the manually classified audio samples that belong to one of the three distinct classes of speech, music, and environmental sound (refer to Chapter 7 further details).

The inputs for each unit is feature vector for F numbers of frames, each with length L. The training frames have $F/2$ frames belonging to the target class, and $F/2$ frames that belong to the other remaining classes. The inputs for the training module are therefore an $F \times L$ feature matrix, and a target class vector of length F. After preparing the training feature matrices, the rows are randomised; this improves the training performance of the classification module.

The outputs of each unit are the trained classification module parameters, that will be used in the classification unit as in Figure 29 shows.

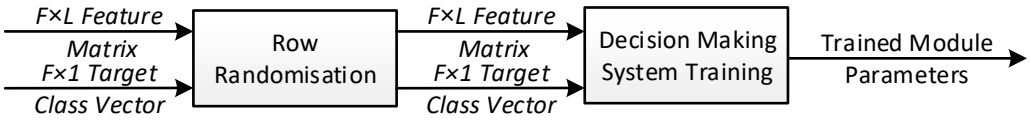


Figure 29: A single class training unit

The classification module is trained to produce the result "1" if the frame belongs to the same class as the classification module, and "-1" if the frame belongs to any other class.

6.7 Detection Pattern Smoothing

The task here is to smooth the classification result of each classification module individually by introducing an average smoothing window of a specific length. The input is the class detection pattern that contains the values of -1 and +1. The value -1 means that the frame does not belong to the class, while the number +1 means that the frame does belong to the class. This pattern is processed by an averaging window and the result will have a real-number value ranging from -1 to +1. The formula in equation 74 is applied to set the pattern to one of the following three values: -1, +1 and 0. Now the result -1 means that the frame does not belong to the class, while +1 means that the frame does belong to the class, and 0 means that the classification result is unknown due to high fluctuation in frames classification pattern.

$$y = \begin{cases} 1 & if\ x \geq T \\ 0 & if\ x < |T| \\ -1 & if\ x \leq T \end{cases} \qquad 74$$

where T is a threshold value that has a value between 0 and 1.

Therefore, if T is set to 0.5, then the result is assigned to +1 if at least half of the input samples have the value of +1, result is assigned to -1 if at least half of the input samples have the value of -1 and the result 0 is set to the remaining cases. The value of 0 indicates one of the following two cases:

a) Transition to another class content. This can be easily overcome by introducing class boundary detection as a post-processing stage. The class boundary detection is not implemented as it is out of the scope of this research.

b) The classification result produced by the classifier was not stable over the averaging window size. That means that the class detector result shows an uncertainty in frame contents within the averaging window; the result was therefore reported as an undetermined class.

Testing results suggest a threshold value of 0.25 in combination with a smoothing window of size 15 frames. These parameters found to provide a good combination that improves the classification performance.

6.8 Collective Decision Making

The task of the collective decision making is to take class detection pattern from the *N* available detectors and combine them into one classification pattern of *N* classes. If the input file contains an *F* number of frames, and it needs to be classified into three classes, the input for the collective decision making are therefore the three vectors in which each has *F* number of smoothed detection patterns of the values (-1, 0, +1). The output is a single vector of length *F* that contains the numbers 1 to *C*, representing the final classification result. The collective decision making is discussed in detail in Chapter 8.

6.9 Classification System Architecture

After the subset feature selection and the training of the *N* classification modules and the trading of the collective decision-making, all these trained modules can be redeployed to be used for testing the intended audio classification task. The building blocks of the

classification phase are illustrated in Figure 30. Each of these building blocks is discussed here briefly:

1. Feature extraction: the input of this stage is an audio file with unknown class content, and the list of feature's subset that is used for classification, see Figure 30. This includes the same steps of normalisation, and framing to prepare the signal for frame features extraction that was discussed before.
2. Frame class detectors: This stage contains *C* detectors for the *C* number of classes, each detector takes the trained module parameter from its corresponding class trained module, the output is *C* number of classes detection pattern of the input audio file.
3. Detection pattern smoothing: This stage produces a smoother version of the class detection patterns to prepare it for the next stage of collective decision making stage.
4. Collective decision making: This is the last stage that takes the smoothed detection parent from the *C* class detectors from the previous stage and uses the trained collective decision module parameters to produce the final classification pattern. See Figure 30.

The details of these stages are similar to the high-level audio content classification and feature selection system that was discussed in sections 6.2-6.7, with the exception that no training is required for decision making systems. Instead, the trained module parameters illustrated in Figure 26 are imported to the corresponding modules in Figure 30 in order to perform the classification directly.

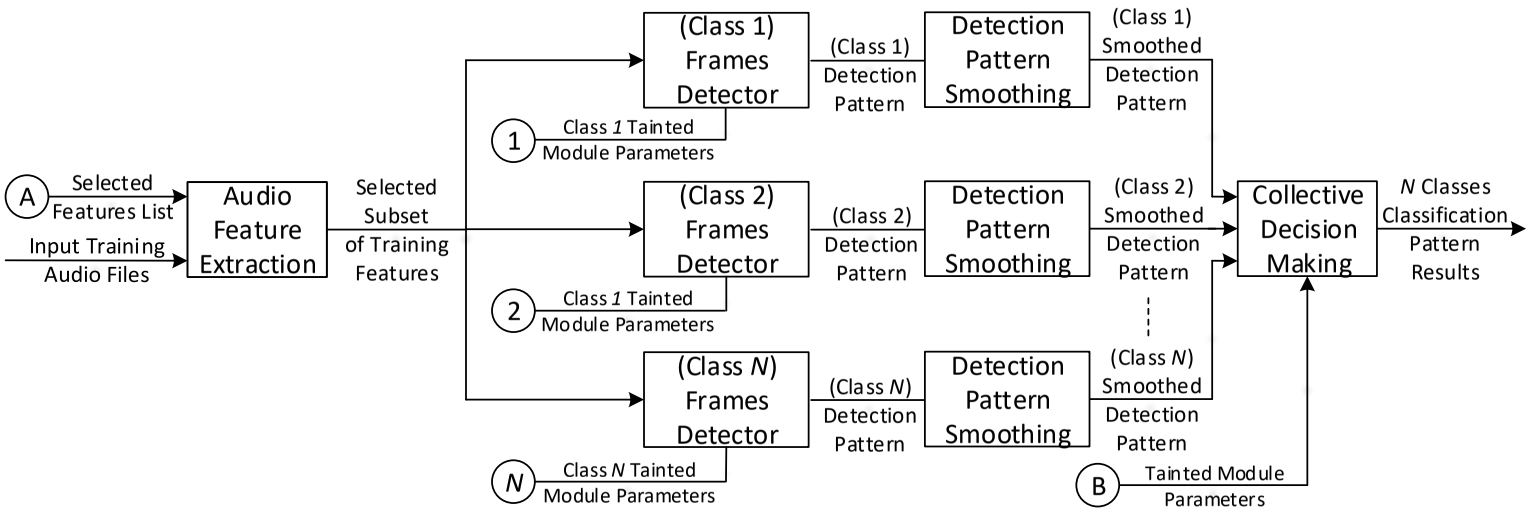


Figure 30: Block diagram showing the architecture of the classification phase.

Chapter 7: Audio Sample Database

7.1. Introduction

The previous chapters discuss the architecture of the high-level audio content classification. This chapter, however, discusses the audio sample database that will be used in chapter 9 for training/testing and validation illustrating the source of the samples and statistics of the utilised databases.

7.2 The Importance of Using a Representative Audio Sample Database

The literature establishes that audio content classification has been a hot research topic since the nineties (Pfeiffer, Fischer, and Effelsberg, 1996; Wold et al., 1996; Scheirer and Slaney, 1997; Delgado-Contreras et al., 2014a; Dhanalakshmi et al., 2010; Phan et al., 2017). There has been a rich development in many aspects of the classification techniques, starting from feature selection, to classification module type and the target classification classes. Apparently, researchers are trying to achieve some improvement in classification performance, but the main problem lies in the lack of a standard audio database on which the performance of different systems can be tested and compared. Yet unfortunately, this is not the case. Each researcher uses his own audio database to test and evaluate his technique, which makes it extremely difficult to compare the classification results produced by different researchers in the field.

7.3 Building the Training/Testing and the Validation Audio Database

The aim of this research is to develop an automated feature selection for high-level audio classification. The high-level classification will process audio samples that exclusively belong to these classes:

1. Speech: these samples contain pure speech with no background music or noise. The speech class has samples of males, females, children, and a group of speakers. The speech samples also cover different emotions such as anger, happiness, and normal speech. The files contain a variety of speech types such as lecture, conversation, shouting, and narration.
2. Music: these samples contain various instruments and music types, genres, and moods.
3. Environmental sound: these samples contain audio samples that do not fit in the previous two classes. Such as the sound of rain, storms, thunder, screaming, a flying helicopter, crashing, a busy road, a schoolyard, clearing throat, coughing, human laughter, door knock, door slam, drawer, keyboard, keys handing, page turning, phone ringing and many other such samples.

No official database could be found that contains such diverse classes content, with sufficiently general samples to be suitable for the intended classification task. Therefore, a new audio database was built by manually collecting audio samples and combining with samples from the available audio databases that are designed to have samples belonging to a specific class only.

The required audio samples should be of high-quality and be a pure class sample. Also, the database should be large enough, varied, and non-biased in order to effectively mimic a real-life audio content; this is a key factor in achieving a successful, practical and truthful result during testing and evaluation.

7.4 Audio File Format and Quality

The audio samples are saved in a WAV file that makes it easier and faster to manipulate. This would avoid introducing any quality degradation or information loss in the

audio data. All the samples are saved with CD audio quality, 44.1 kHz sampling rate and 16-bit depth.

Samples sources should be high-quality files and the audio source is preferred to be in a raw "uncompressed" audio format. Due to the limitations of these files, a very high quality format was also accepted such as the AC-3 format that is used to save the audio in DVD and the DVB-S format used in digital satellite radio broadcasting. These were the only formats that were accepted as a source for the audio database.

The use of such high-quality source files are intended to eliminate any degradation in sample quality and avoid the introduction of artificial differences that might mislead the classification system.

7.5 The Available Audio Sample Databases

There are a couple of audio databases that are available online for researchers for testing and comparison purposes. Some of these databases are:

1. Conference challenge databases; these databases are prepared as a material for audio description contest conferences. In these conferences, participants are invited to submit their work that aims to process the given audio files. Most of these databases are dedicated for scene analysis or music genre classification as shown in the next section;
2. Researcher created databases; they are made available online for the research community like the GTZAN database, the copyright issue is a major reason behind the shortage of such databases online;
3. Sound effect databases; they are created for audio/video editors, an example of these databases is the BBC audio database. The main difficulty in using such databases lies

in the high price tag and the copyright issue that prevent the re-distribution of the database and making them available for research purpose;

4. The benchmark databases; these databases are set as a benchmark for a specific task like speech recognition and music genre classification. Yet unfortunately there is no database available for a high-level audio content classification task;
5. The freely available audio samples; these audio files are contributed by the users, the "freesound.org" website is an example such a source. However, the difficulty is that audio quality and recording conditions used to record these samples are not the same for the audio files achieved from these sources;
6. Music datasets; they contain metadata and pre-extracted audio features of a music tracks, a sample of such databases is the "million song dataset", so these datasets does not provide the original music tracks but their pre extracted features only, the difficulty is that the extracted features may vary, even for the same feature it is likely to use different parameters during feature extraction of utilising a different definition for the same feature between different databases, therefore this source was also omitted.

7.6 The Utilised Audio Sample Databases

Due to the absence of a single database that provides a representative audio sample of the studied classes of speech, music and environmental sound, samples from the following audio databases were utilised to create the training/testing database:

1. The annual conference of the International Society for Music Information Retrieval (ISMIR): The database contains few music tracks that are publicly available. These samples where utilise for validation in Chapter 9 (ISMIR, 2015);

2. The GTZAN genre collection database: This database was created and used by (Tzanetakis and Cook, 1999; Tzanetakis and Cook, 2002). The files in this database were collected from various sources, including CD, radio and microphone recordings. The intended task is music genre detection with a pre-processing stage of speech/music discrimination. The GTZAN audio database consists of 120 audio tracks of a length equal to 30 seconds, half of the samples are speech samples and the other half is music samples. The file format is mono wave file with 22050Hz sampling rate, 16-bit samples. The sampling rate were upsampled to 44100Hz in order to be consistent with the rest of the utilised audio samples.
3. DCASE2016 challenge database: The DCASE2016 challenge is an official IEEE Audio and Acoustic Signal Processing challenge. This challenge listed 4 tasks each having its own database. The utilised database is the sound event detection in synthetic audio and the goal of this task is to detect the sound events, estimate the starting and ending time and assign a class label to the event. The database provided two sets of samples; the first contains 220 isolated even sounds and some speech samples, and the second contains 18 a synthetic mixtures of the same isolated examples in multiple SNR and various events density conditions. Only the isolated events sound was utilized in order to avoid overlapping between the speech and event samples. The audio files are 44.1kHz, 16-bit mono wave files.
4. The sound events database: This database contains events sound recordings made for research purpose. It contains a variety of objects impacts, scrapes, rolls, and deformations; liquid drip, poured, and splashed. All these samples were recorded under the same conditions with high-quality equipment in order to achieve a virtually no differences in background noise or spectral shaping. The purpose of creating this database is to conduct human perceptual experiments to examine the ability to

understand the environment events through the event sound. And locate the regions in the brain that function during events sound perception. The original file format is stereo wave files with 16-bit samples and 44100Hz sampling rate. These samples were converted to a mono format to be consistent with the rest of the audio sample.

These audio databases will be utilised for building the validation database only.

7.7 Sources of the Manually Collected Samples

Audio samples were collected from many sources. One of these sources was is the audio tracks of DVD disks and digital radio broadcast; which provide an excellent source of high-quality samples. The other source is the audio effect websites that provide a high-quality sound effects (for example, freesfx.co.uk, soundbible.com, audiomicro.com). Many of the environmental sound samples were downloaded from these websites.

7.8 Audio Database Description

Two databases where created; the first is used for training/testing and the second is used for validation purpose only. Each of the resulting audio databases contains samples of three classes speech, music and environmental sound. All of the samples were selected to have a homogenous and exclusive class content. The samples were manually classified into one of the three predefined classes. Table 6 illustrates the two databases showing the source of the samples the number of sample from each source and the average sample length.

Table 6: The utilised audio samples details

Database	Class	Source	Number of Samples	Total Number of Samples	Average Sample Length (Sec)
Training/ Testing	Speech	Manually collected samples	334	334	21.4
	Music	Manually collected samples	333	333	33.9
	Environmental sound	Manually collected samples	333	333	20.1
Validation	Speech	GTZAN	39	39	30
	Music	ISMIR	8	49	35.1
		GTZAN	41		
	Environmental sound	DCASE2016	50	200	9.2
		Sound Events Database	55		

The manually collected samples provided the samples for the training/testing database. On the other hand, in the validation database included a randomly selected sample from the mentioned databases exclusively. The GTZAN database provided the samples for both speech and music classes class. While the ISMIR samples where used to provide music samples. Finally, DCASE2016 and the sound even database samples used as a source for environmental sound samples. The sample was collected randomly from these databases, later on, the files that contain mixed data where omitted, this was the case for some mixed speech and music content in both GTZAN and ISMIR databases, also some speech samples in the DCASE2016 Database. Both of the training and the validation databases provided a rich sample representation for each of the three classes. This claim is confirmed by the result illustrated in Chapter 9.

Chapter 8: RFs Algorithms for High-Level Audio Content Classification & Feature Selection

8.1 Introduction

The manual trial-and-error approach of feature selection that has been examined by Al-Maathidi and Li (2012a,b) was found to be ineffective. The main difficulty arises from the high dimensionality of the feature vector. Moreover, feature importance is highly dependent on the target class, the source and file, the format and the quality.

These variables lead to a vast number of choices that need to be tested, which makes the process of testing impractical to explore manually.

On the other hand, the node information gain that is defined in equation 61 can be utilised to predict the feature importance inside the classification tree. The technique of finding the importance of each feature utilised in the classification tree is achieved by finding the total information gain for all the nodes that used this specific feature in the tree. This technique is discussed in detail in this chapter.

8.2 RFs Feature Importance Ranking and Selection

The task of feature selection is to reduce the feature space dimensionality by selecting a subset of features that captures the significant audio properties for the intended classification task. An effective way to achieve an efficient feature selection comes from introducing feature importance ranking for utilised classification features. One way to achieve that is by measuring the features contribution to the final classification result. Therefore, finding an automated feature selection technique that is independent of all of the mention variables is highly important to be utilised for efficient feature selection for classification.

The idea is to utilise an efficient and well-trained classification module and use it to rank the features according to their importance. The training process has some indirect feature selection that is embedded inside a trained classification module. For this task, tree classifiers

are good because they allow checking each and every parent node in the tree using information gain (equation 61). This measure feature purity changes from the parent node to the child nodes. The performance of this technique would be enhanced if an ensemble tree is utilised because the ensemble tree provides an improved classification performance over tree classier.

8.3 Ensemble Tree Feature selection

As discussed in the literature, the ensemble tree feature selection has been used in several fields of pattern recognition. However, the topic of utilising the ensemble tree classifier for high-level audio feature selection has not been given enough attention, although it has many advantages that can be summarised in the following points:

1. Computationally efficient: The ensemble tree technique provides feature selection, not feature dimensionality reduction, so that only the selected subset of features needs to be extracted during classification (while in feature dimensionality reduction all of the features must be extracted before it can be transformed to feature space with lower dimensionality). Allowing only a subset of the features to be extracted during the classification phase. This, therefore, saves time and computational power in both the feature extraction and the training on the reduced feature set.
2. Automatic: The feature selection can be easily automated. The only required manual interaction is to set a single parameter to the accepted tolerance in classification performance degradation. This is discussed in detail in the next section.
3. Classification task related: Feature selection is related directly to the individual classification problem. Thus feature ranking and feature selection are expected to vary for a different classification task even if the frame training features were used.

4. Reliable: The ensemble tree feature selection gives a reliable indicator of feature importance. This performance of feature selection is inherited from the successful classification performance of the ensemble tree. Fortunately, the ensemble tree is one of the most successful classification methods available as the results in Chapter 9 show.
5. Classification-module independent: The classification using ensemble tree might take a longer time compared to other classification techniques such as NNet. This is because the ensemble tree combines the results of thousands of trees and uses a voting technique to produce the final result. After feature selection using the ensemble tree, it is possible to train another classification module that requires fewer resources and lower computational power.

Despite the advantages of ensemble tree feature selection, it comes with one disadvantage. The ensemble trees have no ability to identify redundant features, although it does an excellent job of filtering out irrelevant and the weakly relevant features. This topic will be discussed in detail later in this chapter.

8.4 The Proposed Feature Selection Technique

The aim of a successful features selection technique is to determine the features to be used for classification, and the ones to be excluded. Having a fully automated technique to perform feature selection without compromising the classification performance is highly important.

The adopted approach in this thesis is performed in two steps. The first is the feature importance ranking, and the second is performed by removing the feature that has the lowest importance ranking at each iteration and checking the classification performance using the reduced feature space. Once the performance drops below a pre-defined threshold compared to the training using all the given features, then feature reduction is stopped. The selected

feature set includes all the features used to train the classification module in the previous iteration of feature reduction.

1. Initial training: Train the ensemble tree using all of the available features in order to rank them according to their importance.
2. Performance evaluation: Determine the classification performance of the trained classification module by setting the percentage of positively classified frames to the variable α.
3. Feature importance calculation: Calculate each feature importance, and sort the features list according to their importance.
4. Feature reduction: Reduce the number of features one-by-one using iteration, starting from the least important feature and repeat steps 4.1 and 4.2 while α-β is smaller than the threshold value t.

   4.1 Reduced feature training: Train the classification module using the reduced feature set; at each iteration, the feature-space dimension will be reduced by one.

   4.2 Performance evaluation: Determine the classification performance of the trained ensemble tree that uses the reduced features list for training. In this step, the percentage of positively classified frames is set to the variable β, where β represents the classification accuracy of the reduced feature ensemble tree.

5. Produce the selected features list: Select the optimum features list by including the list of features used in step 4.2 in addition to the feature that was reduced from the list in the previous iteration which caused the performance to drop below the threshold level.

8.5 Training Phase of Ensemble Tree Feature Selection

In order to find the feature importance using the ensemble tree approach, first the ensemble tree needs to be trained using all the available features. After that, it becomes possible to calculate the feature importance. Once the feature importance is calculated, it is possible to select a subset of the features and use them to train any supervised machine learning module including ensemble tree, NNet, GMM. The results are found to be highly promising, even when using only 20% of the features, the classification accuracy remained almost identical (Al-Maathidi and Li, 2015).

This process of feature selection and training with the reduced features is summarised in the following steps:

1. Frame feature extraction: All the features are extracted from training frames. If there is an N number of training frames and a feature vector of size F is extracted, the feature matrix size will be an $N \times F$, also an $N \times 1$ class labels vector will be generated in order to perform the training.
2. Ensemble tree training: The ensemble tree is trained using the $N \times F$ feature matrix.
3. Feature importance calculation: Calculate the importance of each feature in the ensemble trees is calculated by checking how the purity changes between the parent node and the child nodes using equation 61. The result will be a vector of length F that ranks the features according to their importance.
4. Subset feature selection: A features subset of size f is selected, where $f <= F$, to create a feature matrix of size $N \times f$ to be used for training.
5. Classification module training: Training is concluded using $N \times f$ features only. The classification module can be an ensemble tree or any other classification module.

6. Saving trained module and parameters: The trained classification module are saved along with the list of selected features and any other required parameters to be used in classification phase.

Figure 31 shows the block diagram of the ensemble tree feature selection and classification module training.

8.6 Classification Using Ensemble Tree Reduced Features

The following two steps are applied for classification using the trained classification module using a subset of features:

1. Frame feature extraction: The list of f selected classification features is imported from the training phase to train the RFs. Only these features are extracted from the input frames. Therefore, if there is an N frames that needs to be classified, the size of the feature matrix will be $N \times f$;
2. Frame classification: The $N \times f$ feature matrix is classified using the trained classification module from the training phase. The result is an $N \times 1$ classification pattern that represents the class label for the input frames.

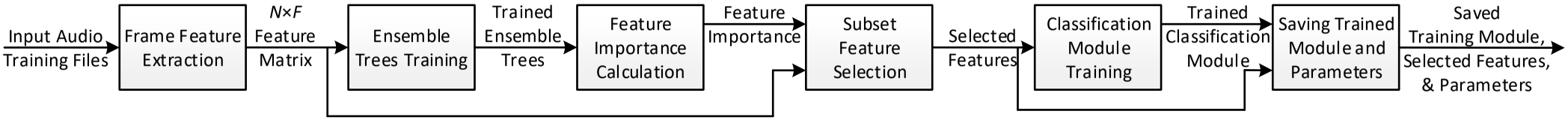


Figure 31: Feature selection using ensemble tree and classification module training

8.7 Collective Frames Classification Decision

To achieve one single classification result for the classified frames, a technique is designed to collect the class detection pattern from the three classes detectors to produce single frame classification result. The input for the collective decision making is the classification pattern from each class detector module and the output is a single vector of class detection pattern that contains class label for each frame as Figure 32 shows.

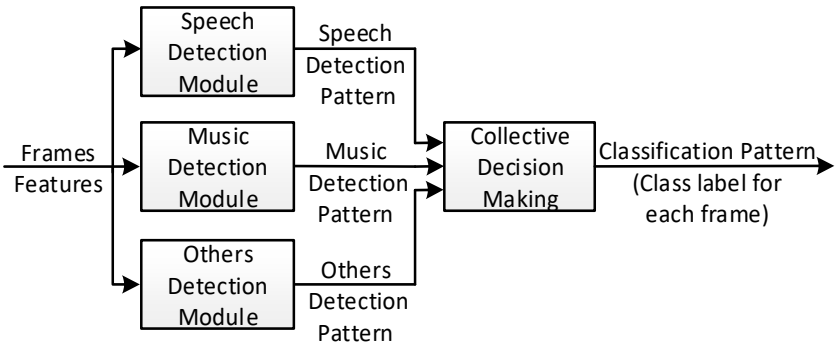


Figure 32: Post processing stage of collective decision making.

The technique was designed to cover a longer time duration (long in comparison with frame length) by combining a sequence of frames classification patterns. This helps to achieve a stable and more reasonable classification result over a long period of time. The case is the same for a human listener who needs to listen for a long enough period to be able to the type of identify audio content.

The required technique must assign a single class label to series of classification patterns of length $n \times c$, where n represents the pattern length and c represent the number of classes to be detected. Each of these patterns has one of the following values "1" if the frame is classified positively, "-1" if the frame is classified negatively, and "0" if the class is unknown. There are two approaches to process this data:

The first approach is to pass all the $n \times c$ patterns to the classification module to classify them into three different classes. This is not a practical approach; it can lead to serious training difficulties since the data is not properly prepared. For example, if n=50 and only one of the frames were wrongly classified, then this can produce 150 different training patterns (50×number of classes). All these patterns should lead to the same classification result because this single frame might occur in any of the fifty possible positions of the three input vectors.

The second approach achieved by splitting the detection pattern into fixed size segments, summarising segment contents by extracting detection pattern statistics and passing it to a multiclass classifier module. The RFs present a suitable option for this task.

The second approach avoids the difficulties occurred in the first approach, therefore this approach is adopted.

The following statistical properties are proposed to summarise the input class detection patterns:

1. Percentage of the values (1, 0, -1): This gives a direct indicator of the amount of frames of each one of the three values, regardless of their position.
2. The standard deviation of the vector: This determines the stability of classification result in the pattern vector.
3. The maximum and the minimum value of the cumulative sum of pattern vector: This indicates the largest number of successive frames that had positively or negatively classified.
4. The range of the values inside the vector: This presents an indicator to the constancy of classification pattern.
5. The interquartile range of the classification result: This indicates the stability of the interquartile classification pattern.

In order to train the collective decision-making module and test its performance, the classification patterns are split into two parts. The first part is used for training, whereas the second part is used for classification. The following steps illustrate the training/classification steps:

1. Initial classification: The audio files are run through the three class detectors; to get the output of three pattern vectors for each file, one pattern from each class detector.
2. Classification vector preparation. The patterns in step 1 are split into equally-sized vectors of size *n* using an *n*-1 overlap between vectors.
3. Pattern splitting: The classification pattern vector is split into two groups: one group is used for training, and the other is used for testing.
4. RFs training: The random forests is trained using the training data from step 3 to map the pattern vectors to the actual class content. Thus the labels "1", "2" and "3" are assigned for the classes, namely *speech*, *music* and *environmental sound* respectively.
5. Vector classification using random forest: The performance of the random forests in detecting the actual class content is tested using the testing vectors from step 3. The result is the percentage for each class in the testing sample that is correctly detected.

The reason to use an *n*-1 overlap is to minimise the delay in the output result production, so once the classification starts the collective decision making module will delay the output result production by time equals to $n \times frame\_length$, so that if the frame length is set to be 40 ms, and *n* set to be 10, then the delay to get the first output from the classifier will be 400 ms. Then, after that, a new classification result will be produced every 40 ms.

Setting a large value to *n* will case a delay in classification result production that make it impractical for real time system, at the same time a smaller value will prevent the collective decision-making system to acquire enough information in order to produce an accurate

classification result. Different values will be tested in Chapter 9 to find the value for this parameter.

Figure 33 summarises the used steps used to combine the frame-classification results. The results in Chapter 9 show that an excellent performance is achieved when this technique is applied.

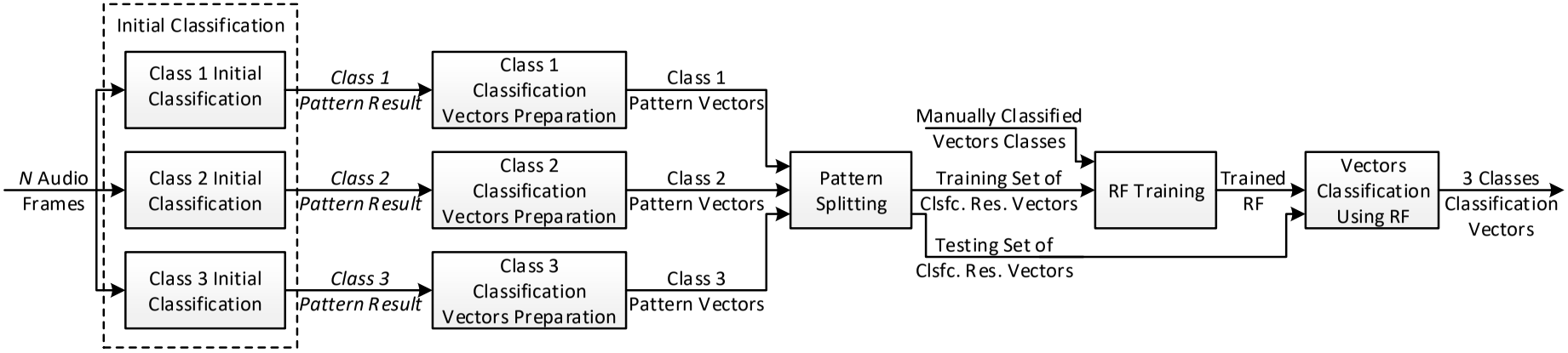


Figure 33: Collective frame classification results

Chapter 9: Experiments and Results

9.1 Introduction

In this chapter, test results are evaluated and discussed to check classification technique performance and feature ranking and selection efficiency. This chapter begins by describing the testing methodology. Later, test results are presented and discussed.

9.2 Test Methodology and Results Presentation

Throughout this chapter, the presented results are subject to a 5-fold cross-validation tests, the results are presented by finding the average and the standard deviation of these tests. It is possible to increase the number of cross-validation folds, but the RFs that represent the heart of the implemented machine learning system has been proven that it cannot overfit (Hastie et al., 2016), as discussed in Chapter 8. This is evident by the low values of standard deviation recorded during testing.

Throughout this chapter, tables are used to present the classification performance. Also, the receiver operating characteristic curve (ROC) is presented to compare the classifiers quality. The rest of this section illustrates some of the terminology and abbreviations that are used.

The numbers in the tables represent the percentage of the correctly classified frames (true classification). For example, if three files contain speech, music and environment sound all sent to speech classifier and the scored results were 90% for the speech file and 80% for the other non-speech files. Then the speech classifier was able to positively classify 90% of the speech frames correctly as speech, and identify 80% of the non-speech frames correctly as non-speech. The result tables also represent the standard deviation over the multiple cross-validation tests.

Another approach was presented to compare the performance of the classifier by showing the ROC values and the area under the resulted ROC curve.

Table 7 lists some abbreviations that will be used in the rest of the chapter.

Table 7: Abbreviations used in the results tables

Abbreviation	Intended Meaning
E	The percentage of truly classified frames that belong to "environmental sound" class
M	The percentage of truly classified frames that belong to "music" class
S	The percentage of true classified frames that belong to the "speech" class
Smth.	The size of the smoothing window
St.Dev.	The standard deviation over multiple cross-validation

All the tests have been carried out using a test tool that was developed using MATLAB. This tool facilitates easy modification of parameters and facilitate the production of multiple test result with a high degree of flexibility; enabling the experimentation of many options through the graphical user interface.

9.3 MPEG-7 Features and Classification Modules Parameters

To extract MPEG-7 features, the proposed features extraction parameters by MPEG-7 standard was adopted as mentioned in Chapter 3, an example of these parameters is the window size (30 ms), hop size (10 ms), ASE size (34 bins). Few other unproposed parameters were tested to be set to the best classification performance. These tests were held using the RFs classifier because it is found to be the most stable classifier that has the ability to produce the best classification results. The parameters selection is discussed in the subsequent subsections.

9.3.1 The Size of ASP Feature

The size of ASP feature vector length and the utilised decomposition algorithm is left open by the MPEG-7 standard. Regarding the length of the ASP vector, it can be as short as 1 or as long as 34 (equal to the size of the ASE vector). At the same time, PCA and SVD both were proposed as a decomposition algorithm by the standard. Therefore, testing was held to check the effect of ASP size and projection algorithm on the classification performance in order to find the optimum classification performance in relation to ASP size. The results in Table 8 show that the best performance was achieved from ASP of size equal to 34 using SVD for projection. Hence, the size of 34 was used in a combination with SVD to extract the ASP feature in the remaining tests.

Table 8: ASP feature size

Decomposition Algorithm	No. of Bins	S	M	E	Av	S St.Dev.	M St.Dev.	E St.Dev.	Av. St.Dev.
PCA	**18**	90.84	85.20	82.26	**86.10**	0.14	0.19	0.18	**0.17**
	22	90.41	84.32	82.68	**85.80**	0.19	0.21	0.33	**0.24**
	26	90.75	83.87	82.52	**85.71**	0.13	0.09	0.18	**0.13**
	30	90.79	84.64	82.78	**86.07**	0.21	0.16	0.24	**0.20**
	34	91.26	84.91	83.55	**86.57**	0.18	0.19	0.13	**0.17**
SVD	**18**	90.25	85.57	84.94	**86.92**	0.30	0.31	0.26	**0.29**
	22	91.23	85.81	85.10	**87.38**	0.25	0.25	0.27	**0.26**
	26	91.74	85.91	84.97	**87.54**	0.25	0.22	0.15	**0.21**
	30	92.07	86.30	85.35	**87.91**	0.33	0.30	0.32	**0.31**
	34	93.46	86.18	85.41	**88.35**	0.39	0.19	0.22	**0.27**

9.3.2 Minimum Frame Power

As discussed before, the use of silent frames to train the classification module leads to training difficulties, because silent frames that will have the same feature vector are expected to appear in the training samples in more than one class. So, before using the frame for training or classification, the normalised audio power N_{Pow} is calculated for each frame in the audio

sample. Later the power of each frame is compared to N_{Pow}, and the frame is used for training/classification only if it exceeds a predefined threshold value.

Testing results of multiple values of minimum frame power were tested and the results presented in Table 9.

The first column in the table represents the threshold value of each test. The results show that the best results were achieved form threshold value equal to 15% therefore, this value set as a default value in the coming tests. No higher values were tested because the high threshold value will lead to ignoring many frames that can contain a useful information for the classifier.

Table 9: Minimum frame energy

Threshold	S	M	E	Av	S St.Dev.	M St.Dev.	E St.Dev.	Av. St.Dev.
0%	90.72	84.51	85.34	**86.86**	0.30	0.37	0.41	**0.36**
5%	92.19	85.18	85.25	**87.54**	0.45	0.20	0.18	**0.28**
10%	93.46	86.18	85.41	**88.35**	0.39	0.19	0.22	**0.27**
15%	94.08	87.15	85.64	**88.96**	0.25	0.17	0.20	**0.21**
20%	93.90	87.71	86.79	**89.47**	0.24	0.21	0.24	**0.23**

9.3.3 The Number of Utilised Training Files

The task here is to find how many training files are sufficient for the evaluation of the classification performance, and to check whether the training/testing database provides a suitable amount audio data to enable the classifier to capture the distinct features of each class. The total number of files in the audio database is 1000 files, and the use of a 5-fold cross validation will make 800 files available for training and 200 for testing. Therefore, the test was started with 200 files and increased gradually to 150 files to reach 800. Classification performance for these tests is presented in Table 10. The classification results were stable once throughout the test, but it shows slight increment in classification accuracy of the first three tests. This proves that the utilised audio sample database presented an efficient sample set to

enable the evaluation of the intended classification task. The remaining test will utilise the full set of the available audio files.

Table 10: Training and testing files counts

No. of Training Files	S	M	E	Av	S St.Dev.	M St.Dev.	E St.Dev.	Av. St.Dev.
200	93.74	86.95	86.14	**88.94**	0.36	0.38	0.18	**0.31**
350	93.82	87.17	86.04	**89.01**	0.53	0.25	0.20	**0.32**
500	94.13	87.19	85.99	**89.10**	0.22	0.20	0.21	**0.21**
650	94.21	87.07	85.77	**89.02**	0.24	0.16	0.21	**0.20**
800	94.08	87.15	85.64	**88.96**	0.25	0.17	0.20	**0.21**

9.4 Machine Learning Technique Parameter Selection

This section discusses the key parameter selection for the examined machine learning technique. These tests were held to fine-tune the classification module to achieve the best possible performance. This will allow better evaluation for the proposed feature selection technique in later stages. The parameter of the machine learning technique will be discussed in one of the following subsections.

9.4.1 Gaussian Mixture Model Parameters

One of the most important parameters in the GMM module is the number of mixtures. With a small number of Gaussian mixtures, the module will not be able to fit the pattern. A very high number might lead to an over-fitting problem. Both cases will affect the classification performance. The classification performance using various numbers of mixtures were tested and results shown in Table 11.

Table 11: Number of Gaussians in GMM

No. of Gaussians	S	M	E	Av	S St.Dev.	M St.Dev.	E St.Dev.	Av. St.Dev.
5	81.3	73.7	72.17	**81.3**	2.1	2.31	1.58	**2.00**
10	83.86	78.44	78.57	**83.86**	1.09	2.67	1.7	**1.82**
15	83.35	74.45	78.92	**83.35**	1.81	1.88	1.49	**1.73**
20	83.31	73.28	78.68	**83.31**	1.48	2.79	2.79	**2.35**
25	82.99	73.66	76.97	**82.99**	2.24	1.9	1.9	**2.01**

The best classification performance was achieved from 10 mixtures, so 10 is the default value for the remaining tests.

9.4.2 Neural Network Parameters

The key parameter in setting a NNet is the number of hidden layers. Table 12 presents the result of multiple tests that used a different number of hidden layers. The result shows that best classification performance was achieved from the NNet with 3 hidden layers, the higher number of hidden layers were not able to improve the classification accuracy, therefore; a NNet with 3 hidden layers is used in the upcoming tests.

Table 12: The number of hidden layers in NNet

No. of Hidden Layers	S	M	E	Av	S St.Dev.	M St.Dev.	E St.Dev.	Av. St.Dev.
1	90.00	76.79	76.94	**81.24**	0.53	1.05	2.71	**1.43**
2	88.26	82.49	81.88	**84.21**	0.37	2.34	1.89	**1.53**
3	87.71	80.36	82.12	**83.40**	1.15	1.92	2.74	**1.94**
4	89.12	76.76	80.77	**82.22**	1.11	0.96	1.84	**1.31**
5	87.84	76.17	80.4	**81.47**	1.43	2.20	3.66	**2.43**

Regarding the neuron transfer function, the sigmoid function used as a transfer function for the hidden layer of the network.

9.4.3 Bagged Tree and Random Forests Parameters

The major parameter here is the number of trees. Multiple tests are performed to find the optimum number. Table 13 and Table 14 present the results of BT and RFs accordingly.

Table 13: The number of trees in the BT

No. of BT	S	M	E	Av	S St.Dev.	M St.Dev.	E St.Dev.	Av. St.Dev.
50	89.75	84.57	82.51	**85.61**	1.20	0.92	0.98	**1.03**
100	91.15	84.86	82.77	**86.26**	1.93	1.05	0.75	**1.24**
200	92.35	85.46	83.77	**87.19**	0.82	1.08	1.17	**1.02**
300	91.13	84.98	83.86	**86.66**	1.74	0.73	0.99	**1.15**
400	91.86	85.20	83.78	**86.95**	0.81	1.11	1.20	**1.04**

The best performance is achieved from the BT while using 200 trees. After exceeding 200 trees, no improvement was achieved. Therefore, 200 trees were set as the default number of trees in the remaining tests.

Table 14: The number of trees in the RFs

No. of Trees in RFs	S	M	E	Av	S St.Dev.	M St.Dev.	E St.Dev.	Av. St.Dev.
250	94.02	86.92	85.55	**88.83**	0.41	0.22	0.31	**0.31**
500	94.29	87.24	85.64	**89.06**	0.20	0.18	0.23	**0.20**
1000	94.21	87.15	85.83	**89.06**	0.13	0.13	0.11	**0.15**
2000	94.37	87.18	85.72	**89.09**	0.09	0.12	0.13	**0.11**
4000	94.19	87.25	85.78	**89.07**	0.12	0.07	0.10	**0.10**

In the RFs case, the best performance is achieved by using 2000 trees, with a very small improvement over the smaller forest sizes. In the upcoming tests, the number of trees in the forest is set to 1000; this will speed up the result production without having any major degradation in the classification accuracy.

Even though the RFs continued a higher number of trees it was much faster than the BT and the results of RFs show an improvement over the BT classification accuracy. This improvements does highlight benefits of using the bootstrapping technique.

9.5 Comparing the Performance of Classification Modules

After selecting the optimum parameter for each module, the classification performance is compared for the mentioned classifiers. The results presented also the effect of utilising the step of classification pattern smoothing, that introduced to improve the classification performance.

The classification performance of GMM, NNet, BT and RFs is compared in Table 15. The shown result represents the best performance of each module using all of the extracted

MPEG-7 features, the total size of frame feature is equal to 49 (34 of them belong to ASP).

The resulting classification pattern of each module is smoothed with a smoothing window of the size 15, 25, and 50 frames.

Table 15: Classification module performance using 49 features

Classification Module	Smth.	S	M	E	Av	S St.Dev.	M St.Dev.	E St.Dev.	Av. St.Dev.
GMM	**1**	83.86	78.57	78.44	80.29	1.09	2.67	1.70	1.82
	15	88.15	82.68	83.08	84.64	1.15	2.88	1.82	1.95
	25	88.64	83.17	83.69	85.17	1.17	2.90	1.84	1.97
	50	88.88	83.53	84.13	85.51	1.17	2.94	1.85	1.99
NNet	**1**	88.26	82.49	81.88	84.21	0.37	2.34	1.89	1.53
	15	90.97	86.11	86.01	87.70	0.41	2.39	1.85	1.55
	25	91.19	86.40	86.43	88.00	0.40	2.39	1.87	1.55
	50	91.35	86.76	86.96	88.35	0.33	2.47	1.98	1.59
BT	**1**	92.35	85.46	83.77	87.19	0.82	1.08	1.17	1.02
	15	94.75	88.08	85.72	89.51	0.96	1.11	1.29	1.12
	25	94.95	88.43	85.90	89.76	0.97	1.14	1.34	1.15
	50	95.13	88.92	86.18	90.07	0.91	1.12	1.40	1.14
RF	**1**	94.21	87.15	85.83	89.06	0.13	0.13	0.11	0.15
	15	96.34	89.28	87.63	91.08	0.12	0.09	0.10	0.10
	25	96.54	89.56	87.83	91.31	0.12	0.07	0.10	0.10
	50	96.78	89.95	88.23	91.65	0.14	0.09	0.12	0.12

As shown above, the average smoothing window improved the classification result in all the tests; the efficiency was in a linear proportion to the window size as the results presented in Table 15. The performance of these four classifiers is presented using ROC curves in Figure 34; the curve shows the improved classification of the RFs and BT over the GMM and NNet, while Table 16 lists the area under ROC curves.

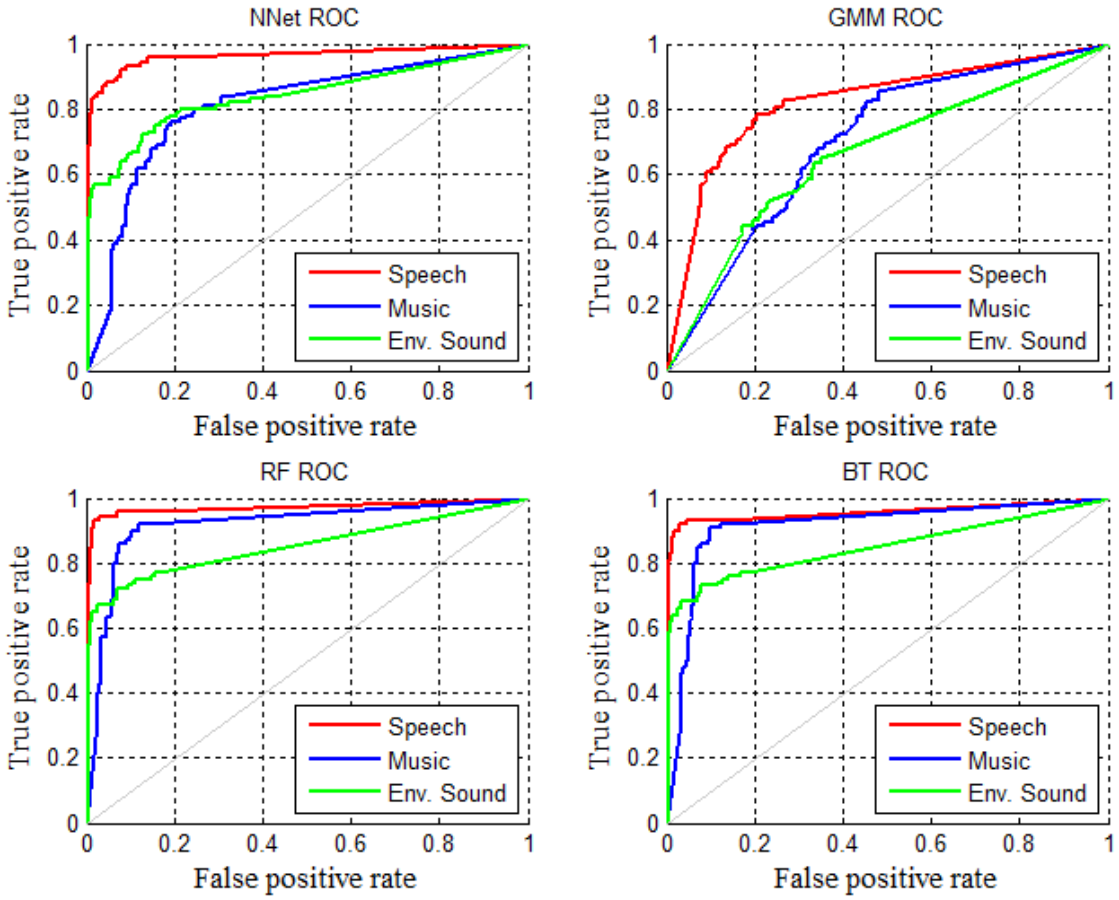


Figure 34: ROC curves for NNet, GMM, RFs and BT.

Table 16: The Area under ROC curves

	NNet	GMM	RFs	BT
Speech	0.96	0.93	0.97	0.96
Music	0.81	0.80	0.92	0.91
Env. Sound	0.84	0.76	0.85	0.85
Average	0.870	0.830	0.913	0.907

The GMM classifier shows the lowest classification accuracy across the three classes scoring average accuracy of 80.29% true classification along with the maximum standard deviation values, while the NNet performed better than GMM and scored 84.21%. The best classification performance archived using the BT and the RFs; the performance was comparable and they scored 87.19% and 89.06% accordingly. The lowest standard deviation was scored by the RFs, indicating that classification performance was stable across the five tests. These results support that the RFs does gain much from the cross validation.

Due to the low classification performance achieved by the GMM, it will not be used for evaluation in the coming tests.

The results illustrate that the best classification performance was achieved in speech detection modules for both positive and negative classification. This is clearly noticeable in the result of Table 15, Table 16 and ROC curves in Figure 34. The case is the same for the listed classification models, it is also visually clear and noticeable from the ROC curves. The result also shows that environment sound detection modules scored the lowest performance in all tested modules except for the NNet. This result is expected since the content of music class has higher variation in content compared to speech class, and the environmental sound has the highest variation in content compared to both speech and music classes. Therefore, the result was the best in speech followed by music then the event sound classes. The next section will put this ranking to the test and check the classification performance.

9.6 RFs Classification Feature Importance List

The importance level of each classification module is calculated using the GINI purity measurement on the trained RFs. Therefore, for the three classification tasks, a separate RFs is utilised for each one of the classes class. The results in Table 17 show how the different classification tasks require a different feature set. This promotes the main advantage of RFs based feature selection link the feature selection to the classification task in contrast with feature selection that is based on features statistical dependency methods. The last column in Table 17 lists the overall importance that is achieved by finding the average of the classes importance.

Table 17: RFs-determined feature importance

Ranking	Speech		Music		Environment Sound		Overall Importance	
	Importance	Feature	Importance	Feature	Importance	Feature	Importance	Feature
1	10.13	ULH	16.72	TC	11.51	ASP 1	10.73	TC
2	8.78	ASS	8.02	ULH	7.38	TC	7.13	ULH
3	8.08	TC	5.46	HSC	6.60	ASC	6.23	ASP 1
4	6.81	ASF	4.22	ASP 1	5.51	ASS	5.00	ASS
5	5.86	AFF	4.10	AP	4.86	HSC	4.32	HSC
6	4.53	HSV	3.51	ASP 2	3.25	ULH	3.85	ASC
7	3.84	ASP 2	3.28	HSS	3.21	HSV	3.23	ASP 2
8	3.45	ASC	2.44	HR	3.16	ASP 3	3.06	AFF
9	2.96	ASP 1	2.35	ASP 3	2.79	LAT	2.93	ASF
10	2.81	HR	2.23	LAT	2.74	AP	2.74	HSV
11	2.69	ASP 5	2.18	AFF	2.68	ASP 4	2.55	AP
12	2.63	HSC	2.00	ASP 5	2.35	ASP 2	2.48	ASP 3
13	2.32	LAT	1.85	MIN	2.24	ASP 5	2.45	LAT
14	1.94	ASP 3	1.78	ASP 4	2.24	ASP 32	2.31	ASP 5
15	1.82	HSS	1.78	ASP 6	1.68	ASP 6	2.27	HR
16	1.82	ASP 34	1.57	ASP 34	1.58	HSD	2.14	HSS
17	1.58	ASP 32	1.53	ASP 12	1.58	HR	1.83	ASP 4
18	1.44	ASP 31	1.49	ASC	1.51	ASP 31	1.58	ASP 6
19	1.28	ASP 6	1.35	MAX	1.49	ASP 29	1.51	ASP 32
20	1.20	ASP 29	1.31	ASP 18	1.47	ASP 9	1.46	ASP 34
21	1.10	ASP 12	1.30	ASP 22	1.44	ASP 16	1.33	ASP 31
22	1.07	ASP 19	1.30	ASP 23	1.42	ASP 33	1.32	ASP 12
23	1.05	ASP 21	1.29	HSD	1.39	ASP 18	1.25	ASP 29
24	1.05	ASP 4	1.27	ASP 10	1.33	HSS	1.23	MIN
25	0.99	ASP 33	1.27	ASP 14	1.32	ASP 12	1.20	ASP 18
26	0.98	ASP 24	1.26	ASP 28	1.25	ASP 10	1.19	ASP 16
27	0.96	ASP 22	1.23	ASP 13	1.22	ASP 19	1.17	HSD
28	0.95	ASP 23	1.23	ASP 16	1.19	ASP 7	1.15	ASP 9
29	0.94	ASP 28	1.22	ASP 27	1.17	ASP 14	1.13	ASP 19
30	0.93	ASP 15	1.22	ASF	1.14	AFF	1.11	ASP 10
31	0.91	ASP 18	1.21	ASP 24	1.13	ASP 15	1.10	ASP 14
32	0.90	ASP 9	1.18	ASP 7	1.13	ASP 8	1.09	ASP 33
33	0.90	ASP 16	1.17	ASP 17	0.99	ASP 34	1.07	ASP 15
34	0.88	ASP 14	1.15	ASP 21	0.98	MIN	1.06	ASP 7
35	0.86	MIN	1.14	ASP 15	0.98	ASP 13	1.06	ASP 22
36	0.82	ASP 7	1.11	ASP 19	0.98	MAX	1.06	ASP 23
37	0.81	ASP 10	1.09	ASP 9	0.96	ASP 17	1.04	ASP 21
38	0.80	AP	1.08	ASP 8	0.93	ASP 21	1.01	ASP 24
39	0.80	ASP 11	1.05	ASP 29	0.93	ASP 11	1.00	ASP 28
40	0.74	ASP 30	1.02	ASP 31	0.92	ASP 23	0.97	ASP 13
41	0.71	ASP 13	1.01	ASP 30	0.92	ASP 22	0.97	ASP 8
42	0.70	ASP 8	0.89	ASP 11	0.83	ASP 24	0.92	MAX
43	0.70	ASP 27	0.88	ASP 20	0.83	ASP 25	0.92	ASP 17
44	0.68	ASP 26	0.86	ASP 33	0.82	ASP 20	0.91	ASP 27
45	0.66	ASP 25	0.79	ASP 26	0.81	ASP 30	0.87	ASP 11
46	0.64	HSD	0.75	ASP 25	0.81	ASP 27	0.85	ASP 30
47	0.63	ASP 17	0.71	ASP 32	0.80	ASP 28	0.75	ASP 26
48	0.44	ASP 20	0.70	ASS	0.78	ASF	0.75	ASP 25
49	0.44	MAX	0.47	HSV	0.78	ASP 26	0.71	ASP 20

The resulted feature importance indicates that the SVD has done a relatively good job in preserving the important information in the first few SVD components. The first 3 ASP components appeared in the top 12 ranked features, and these ASP coefficients appeared in order. The ULH feature was able to capture an important property of both speech and music and this complies with the low harmonic level of speech compared to music. The case is similar for the TC scored high ranking over the three classes that scored the same ranking in all of the three classes. The top ranked spectrum descriptors are the ASS, ASC and ASF. Those three features make it to the top 9 features by being able to discriminate the spectrum shape of the different audio class content.

9.7 The Performance RFs Bases Feature Selection

The overall feature importance listed in Table 17 will be evaluated using wrapper method that utilises both of NNet and RFs as a classification module. Both results are presented in Figure 35.

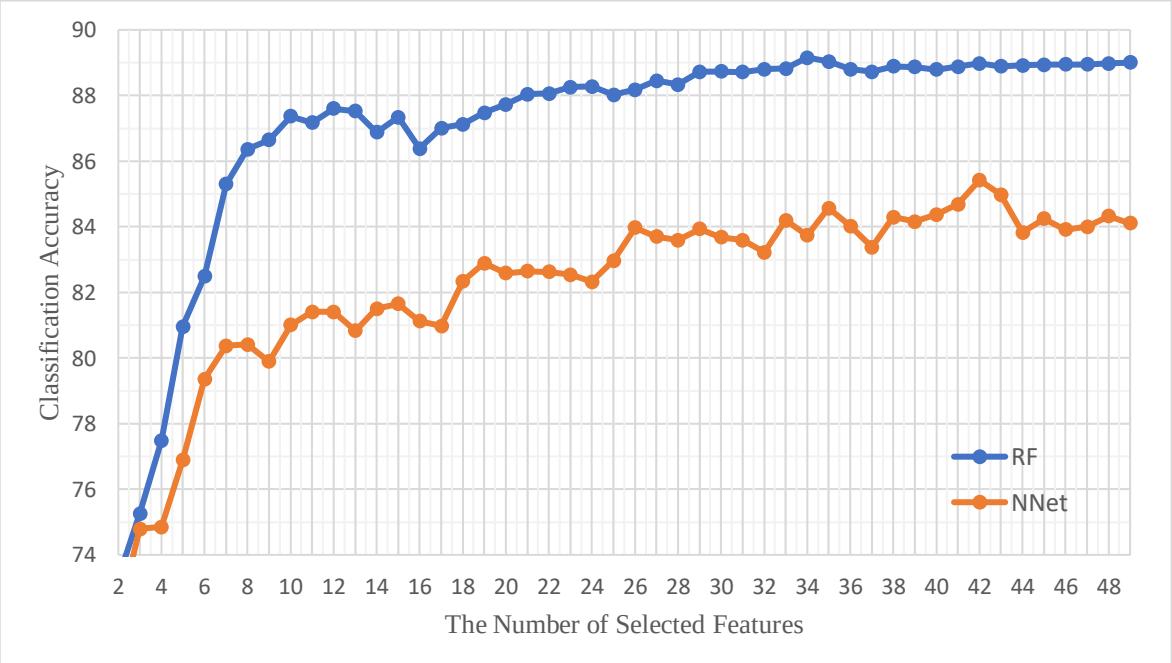


Figure 35: RFs feature selection effect on classifier performance

It is clear that the proposed RFs based feature selection performed very well with both NNet and RFs this proves that feature ranking and selection can be generalised to be used by classifiers other than the RFs.

The reported classification accuracy using all the 49 features was 89.06% for the RFs and 84.21% for the NNet. Figure 35 shows that RFs was able to sustain the classification performance using only 34 features out of 49 (69.4%), while using only 21 features out of 49 (42.9%) caused a performance degradation of 1.17%. Classification performance sustained a good performance by utilising only 10 features (20.4%) with degradation in classification performance of 1.7%. after that the performance started to drop rapidly after keeping less than 4 features, so the classification results nnot 4 features were omitted. The Behaviour of both NNet and RFs is comparable, it sustained the performance with 20 features (40.1%), lowering the classification performance by 0.2% only. After that, the classification performance starts to lower until it reaches a degradation in performance of 3.8% by utilising just 7 features only (14.3%).

MPEG-7 standard has not proposed any feature selection method to compare with, but it did utilise a future dimensionality reduction technique, this includes PCA that projects the data to a lower dimensional system while retaining as much of the information as possible (Kim et al., 2005). The result of PCA feature manipulation will produce a new set of extracted features that have lower dimensionality.

The result of PCA feature extraction and how it affects the RFs classification accuracy is presented in Figure 36. The result shows a drop of (3.42%) in the classification performance after reducing only 6 out of the 49 features. The second big drop appears while utilising 9 features only. At that point, the PCA and RFs combination scored classification accuracy of 77.9%, while the RFs feature selection combined with RFs classifier scored 86.4%.

Although these two techniques are different in some aspects, it is mentioned here just to shed the light on the performance of the RFs feature selection within the MPEG-7 framework.

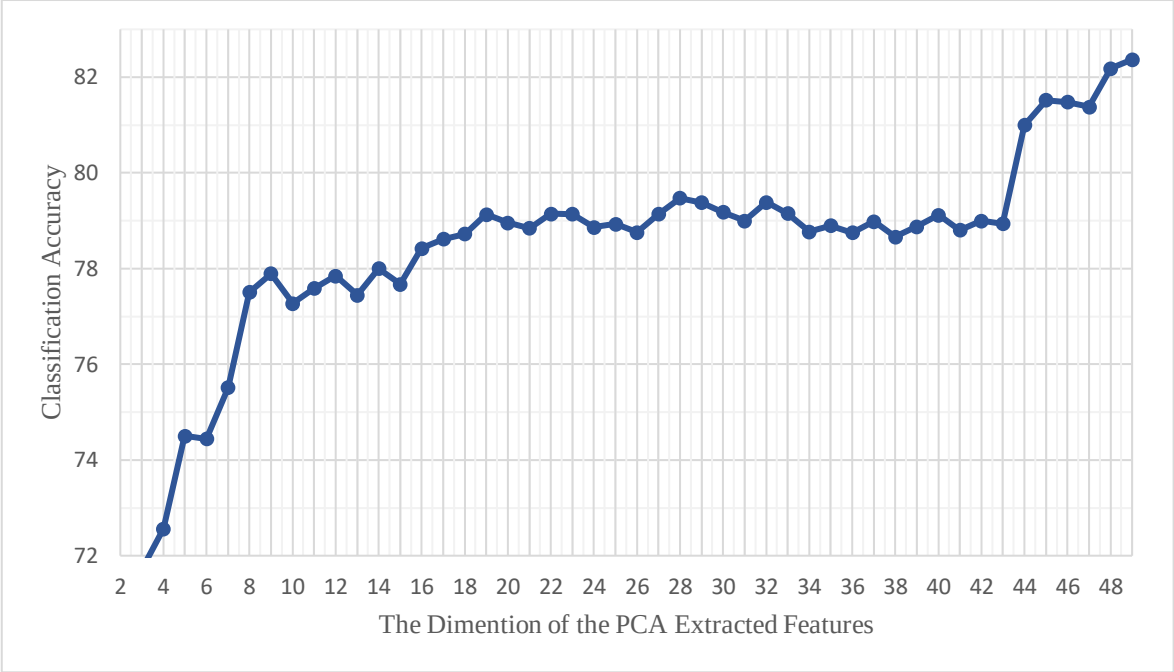


Figure 36: PCA reduced feature extraction and its effect on classification performance

9.8 Collective Decision Making

The final stage of collective decision making is intended to process the class detection patterns that were produced by the three class detectors to produce a single class label for each input frame. The only parameter that needs to be tested is the length of the analysed pattern. Multiple tests are performed and their results are presented in Figure 37.

The highest result scored before introducing the collective decision making was 98.06% using all the 49, while the utilising 10 features only brought the result to 87.73%; both of these values are presented in Figure 37. The collective decision making was able to improve the classification performance dramatically scoring a classification performance better than the highest performance that was achieved by the RFs classifier even with a small size of analysed pattern. The analysis of a larger segment of audio introduced a greater improvement. The case

is the same for humans that usually require keep listening for a short period of time to be able to give a conclusive decision about the type of audio content.

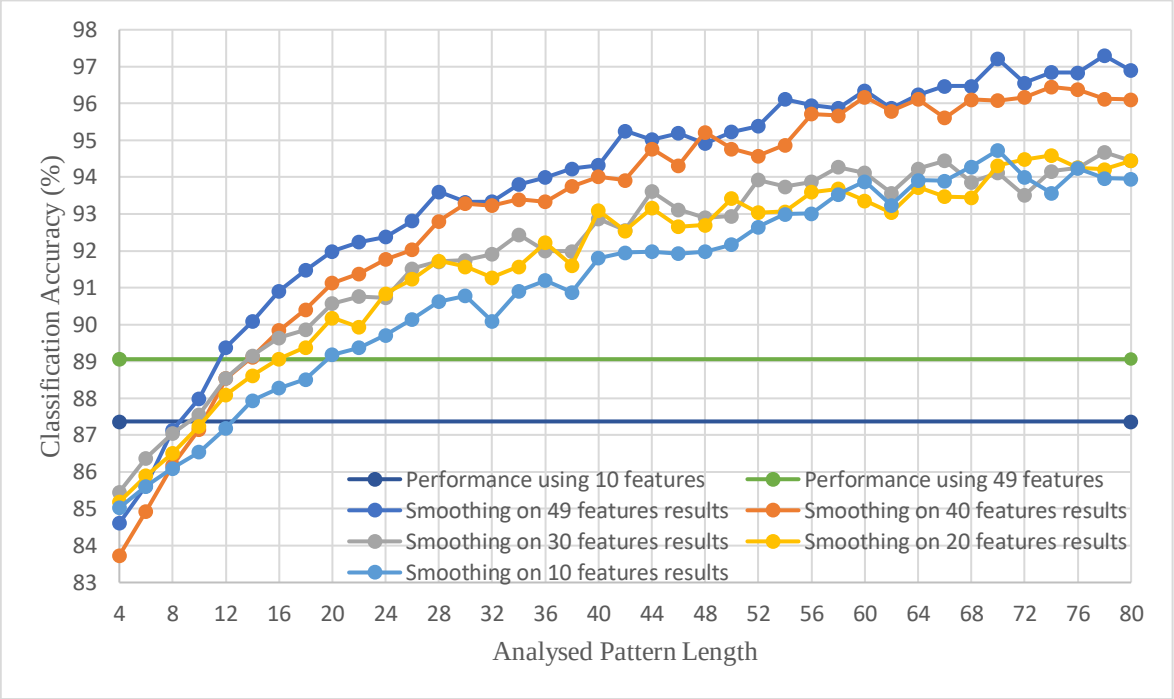


Figure 37: Collective decision making performance

The results show an excellent improvement over the classification pattern of the RFs even when a small analysed pattern was utilised. This stage can cover up any possible degradation might have accrued after feature selection.

9.9 Validation of the Proposed Feature Selection Technique

To validate the previously reported results, a fresh sample set will be introduced to the previously trained classification/feature selection technique. None of these samples or any other samples from the validation databases were used for training before. The number of audio samples and their details were mentioned in Chapter 7.

In the first place, the results of RFs based feature selection is presented to show how the feature selection will affect the classification performance. The figures will present the

result of each validation dataset separately with a single figure for each class. The performance of the testing data will be presented for comparison purpose only.

The classification accuracy of the speech audio samples is presented in Figure 38, followed by music samples in Figure 39 and finally the environment sound in Figure 40.

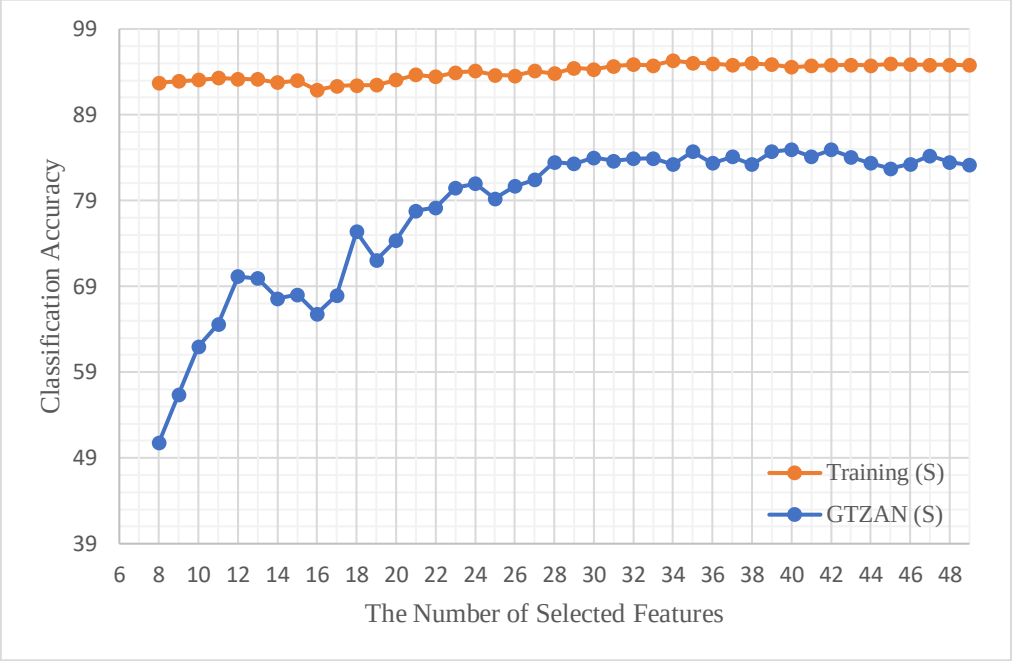


Figure 38: The validation of RFs based feature selection for speech

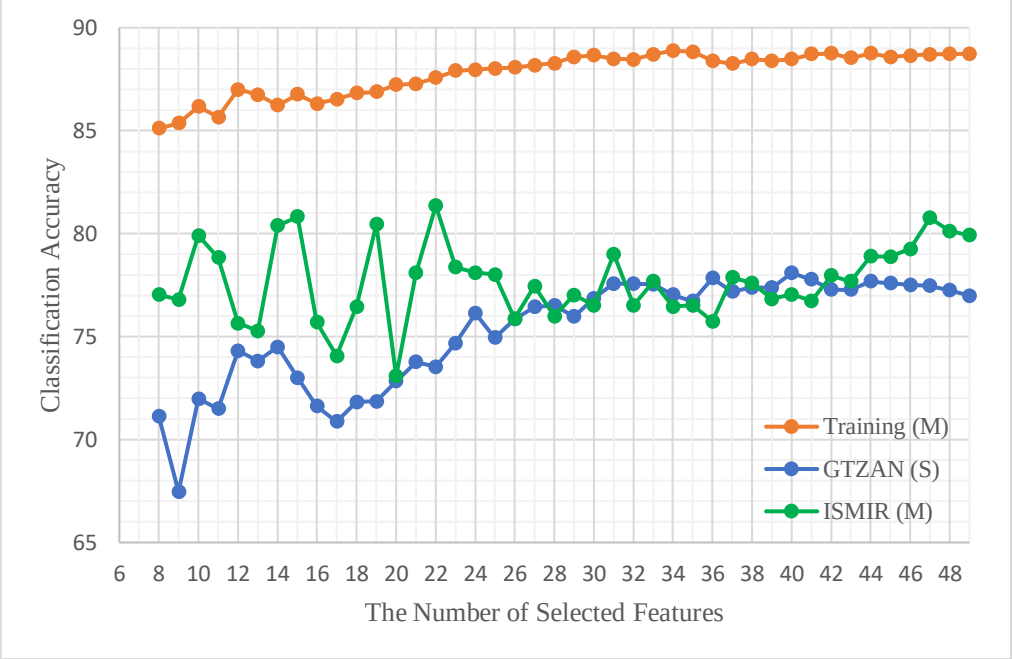


Figure 39: The validation of RFs based feature selection for music

It is noticeable that the classification performance in validation test is lower than the results achieved from the cross-validation test which is expected for validation. The feature selection performed well with only 28 out of the 49 features (57.1%), and show a rapid drop only while using less than 12 features (24.5%). One of the reasons behind the lower classification performance is the low quality of the test samples that were collected from different sources without taking in consideration the quality of the recording and their background noise level. The reported results show a classification performance reduction in validation test that is comparable to the reduction in cross-validation test. The feature selection technique was yet able to exclude the less important classification features proved by the ability of the classifiers to reasonably sustain the classification performance for multiple iterations of feature reduction.

GTZAN dataset in the music class shows almost the same trend as speech samples. The classification performance scored 76.98% using 74 features, and it hit the peak of 78.08 using 10 features and started the rapid drop after reducing the feature to less than 12 features (24.5%). The ISMIR dataset showed some high fluctuation around the average classification performance, and the main reason behind this is the small number of audio samples in this database. Generally speaking, the classification performance started the high fluctuation while using 22 features or less (44.9%).

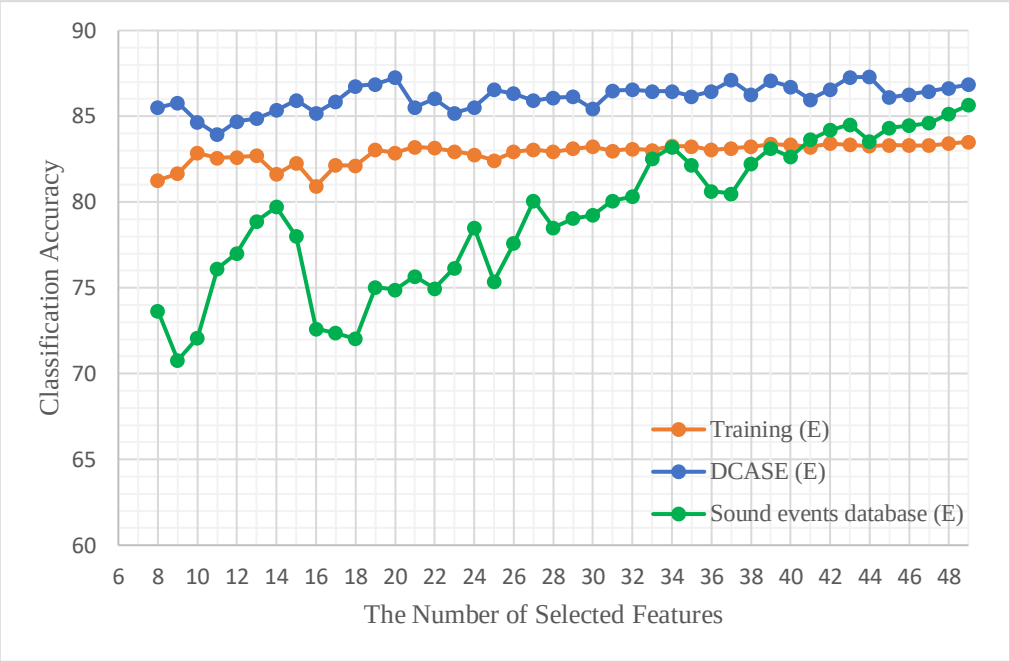


Figure 40: The validation of RFs based feature selection for environment sound

The future selection performance along with classification accuracy showed a great result for DCASE database. The trend line shows a fine match to the training samples, scoring the best classification accuracy of 87.26 using 20 features only (40.8%), and the fast performance degradation started after using 6 samples only. The case was the same for the sound event database samples for the first 10 tests. The highest performance of 85.65% was archived using all of the 49 features, while classification performance stability decreased while using less than 16 (32.7%).

The overall validation performance is presented by averaging the classification results for all the classes in Figure 41.

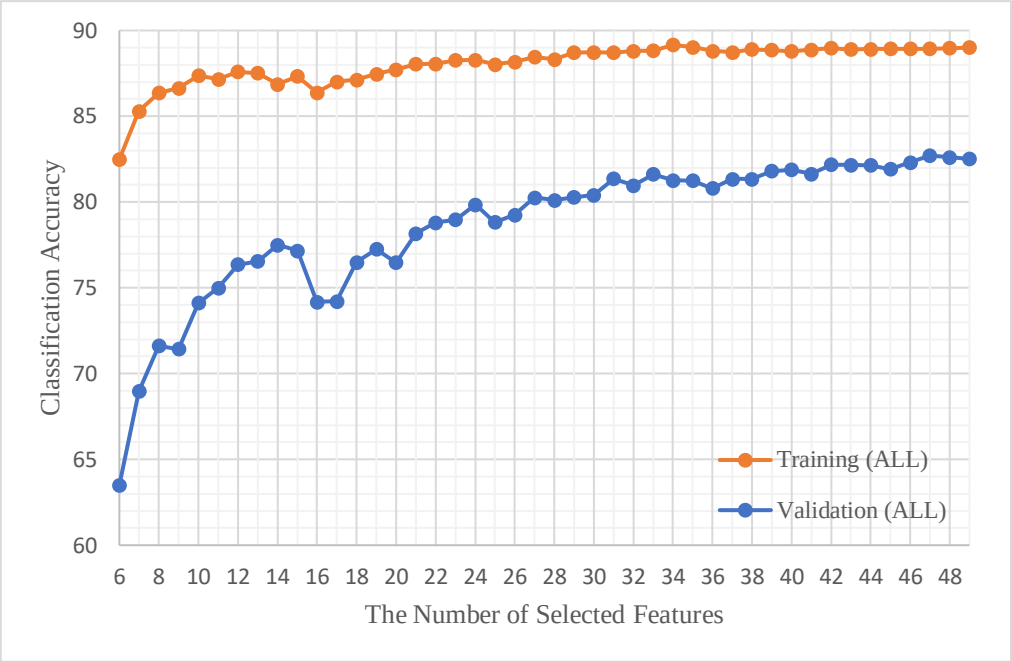


Figure 41: The overall performance of validation of RFs based feature selection

Both of the validation and training data showed almost the same trend training of having their peaks at a large number of features, and started the gradual descent until the number of features reaches 8-10. The highest accuracy achieved from validation samples is 82.7% utilising 46 features while testing result was 89.1% using all of the 49 features. The validation data shows a degradation in the classification performance of 5.2% by utilising only 14 features with feature reduction ratio of 28.6%.

Figure 42 shows the ROC curve for the overall classification performance using 4 different number of features of feature vector length including 49, 36, 23 and 10. These numbers were selected to show the best performance using 49 features and a low yet relatively stable performance using 10 features and two other values in between.

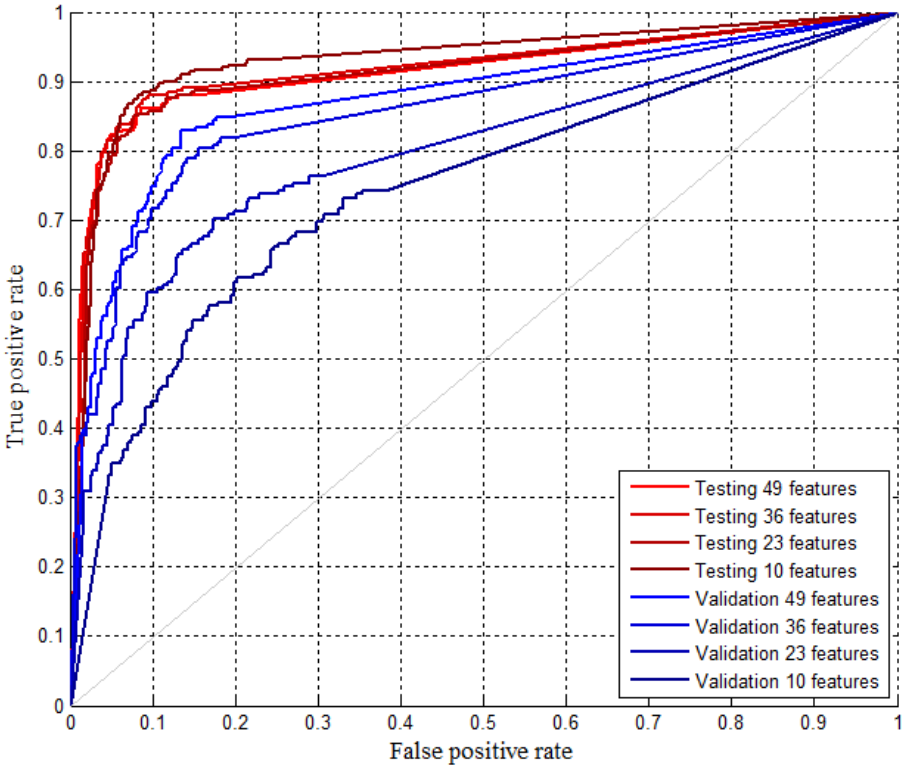


Figure 42: Classification performance using 10, 23, 36 and 49 features.

The figure confirms the previously presented result by showing a higher degradation in classification performance while utilising a small number of features. Still, the RFs based feature selection technique has been proven as an efficient feature selection technique that shows a promising robustness against changes in audio datasets and the ability to perform feature selection for the high-level audio content classification with all of the variation for each of the tested three classes.

To conclude, the presented features ranking provides for the classes of speech, music and event sound present an efficient feature selection guide for other researchers in the field. This list of ranked features that have been tested and validated presents a successful starting point for other researchers in the field.

an efficient features list for each that can be adopted by other researchers for these classes

1. The resulted feature list along with their ranking for the studied classes

9.10 Validation of the Proposed Collective Decision Making

The collective decision making presented a good improvement in the classification data set. Using the validation data set, the result in Figure 43 shows the classification accuracy achieved by collective decision making after processing the classification results that utilised 10, 23, 36 and 49 features. The results show a significant improvement while utilising wide patterns of classification results. This could be complement degradation in classification performance even while utilising a small set of features.

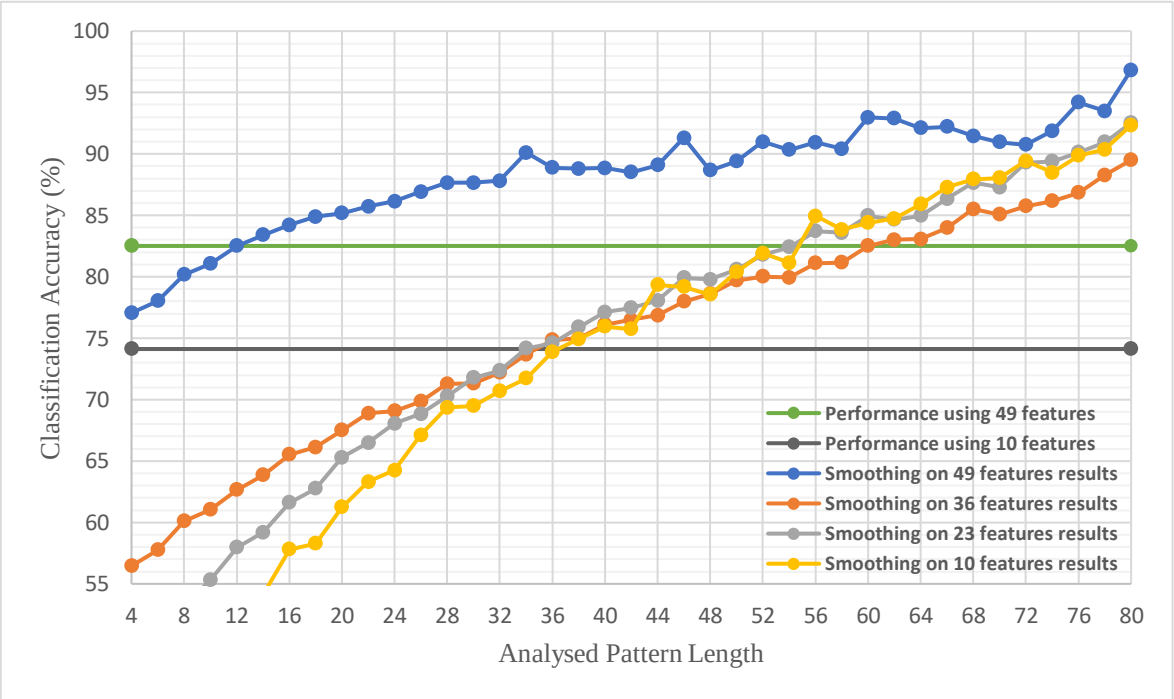


Figure 43: Collective decision making performance on validation data

The highest scored classification result is 82.7%. After applying the smoothing window, the results of all the tests exceeded this threshold level by analysing 60 classification patterns. While widening the analysed pattern size a little, the classification result dramatically improves. The only drawback is the time delay to produce the smoothed results that is caused by this stage. Even though it is not that significant delay but it still might be considered as an issue for real time applications. In that case, a balance need to be kept between the analysed

window size and the utilised number of classification features in order to achieve the required performance.

Chapter 10: Conclusions & Future Work

10.1 Conclusions

The usefulness and importance of high-level audio classification as a pre-processor or input stage for audio information mining are well appreciated and acknowledged. An efficient, automatic and systematic feature selection regime with a suitable classification algorithm can improve classification efficiency and the overall system performance.

This thesis presents a systematic approach for subset feature selection and two stages approach for audio content classification that can efficiently classify audio content into speech, music and environmental sound.

The features ranking and selection part is based on RFs feature importance, as a result of this ranking, a list of features evaluated and proposed for each of the before aforementioned classes. This feature list found to provide an efficient feature ranking not only for the random forest classifier but also for another machine learning classifier, the resulted feature ranking tested using wrapper method with backward search utilising both of NNet and RFs classifiers. The resulted feature ranking was evaluated twice; the first time using cross validation test and the second time it was evaluated using a new different dataset of validation samples (the validation samples have not been presented in the training phase). Both testing and validation results presented the efficiency of the suggested features list for the three mentioned classes this evidenced by the reported results in Chapter 9.

The first stage of audio content classification utilised three binary classifiers; each classifier is trained to detect the presence of one of the aforementioned classes. The reported classification results showed a promising classification performance even after a substantial reduction in the number of the utilised features. The second stage of collective decision making combines the three class detector results to produce a single class label for each audio frame. This stage shows the ability to improve the final classification accuracy significantly and

allows to compensate any possible degradation in classification accuracy after feature selection.

This research conclusions are summarised in the following points:

2. RFs classifiers provide an efficient module for the high-level audio content classification. The main strength of RFs classifier is the ability not to overfit small training dataset that enables the generalisation to other validation datasets. The RFs classifier also showed the ability to provide a stable classification results with a performance better than the other tested classifiers.
3. RFs present an effective systematic tool of feature importance ranking; this presents a better option than feature distance and correlation measurements because it does consider the target classification task to perform feature ranking.
4. The resulted feature list along with their ranking -for the studied classes- present a feature importance guide for other researchers in the field of audio content classification. In other words, the list of ranked features for each aforementioned class presents a successful starting point for other researchers in the field.
5. The dimensionality of the feature space can be reduced dramatically using the suggested technique. For example, in cross-validation test, the RFs based feature selection was able to omit 57.1% of the training features while reducing the classification performance by only 1.17%. For validation, 71.4% of the features were omitted while reducing the classification performance by 5.2% only. This amount of feature reduction will lower the required computational power for both feature extraction and classification time; this will make the technique more feasible for real time application.
6. The selected features can be utilised by classifiers other than RFs, evidenced by the reported NNet performance in Figure 35.

7. The collective decision making provided an excellent post processing stage that smooths the frame classification pattern to the classification result. Moreover, it can cover up any degradation in classification performance that might be introduced after features selection.

The major limitation of this research lies in the utilisation of high quality samples that does not contain any background noise or mixed class content. The samples in GTZAN database presented some lower quality samples; this had slightly affected the classification performance reported in Chapter 9. However, this limitation can be overcome by introducing some lower quality samples to the training set, and this is expected to make the classifier more robust when handling the lower quality audio samples.

10.2 Future Work

From the work done in this thesis, some limitations and possible developments were found that require further study in the future. The main topics that need to be addressed are:

1. Feature ranking and selection for mixed classes content. The implemented feature selection and classification technique could be further developed to handle the overlapped class content, allowing a more practical tool to handle the real-life audio data.
2. Noisy audio content. This work has the limitations of not being able to handle low quality noisy audio data due to the fact that all the training/testing data contains only clean samples. To overcome this limitation, some lower quality samples can be introduced to the training/testing database. Another option is to utilise a pre-processing stage of denoising audio data before using them in the training/classification process.
3. Utilising further stages of lower-level classifiers. This will create a hierarchical approach to classify the processed high-level class into further subclasses. These subclasses will provide a more detailed description that will potentially improve the performance of any further information mining and scene-analysis tools.

4. Developing a new feature set to handle more challenging classification tasks. The implemented feature ranking technique can be used to evaluate their ability to improve the classification results.

References

Al-Maathidi, M. M. and Li, F. F. (2012a) Feature Spaces And Machine Learning Regime For Audio Content Classification And Indexing. *SDIWC International Computer Science Conferences*, 262-274.

Al-Maathidi, M. M. and Li, F. F. (2012b) NNet Based Audio Content Classification and Indexing System. *International Journal of Digital Information and Wireless Communications (IJDIWC)*, 2(4), 66-78.

Al-Maathidi, M. M. and Li, F. F. (2015) Audio Content Feature Selection and Classification, A random Forests and Decision Tree Approach, *International Conference on Progress in Informatics and Computing (PIC-2015)*. Nanjing, China, 18-20 December 2015.

Alías, F., Socoró, J. and Sevillano, X. (2016) A Review of Physical and Perceptual Feature Extraction Techniques for Speech, Music and Environmental Sounds. *Applied Sciences*, 6(5), 143.

Aronowitz, H. (2007) Segmental Modeling for Audio Segmentation, *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. 15-20 April 2007.

Baniya, B. K., Lee, J. and Li, Z. N. (2014) Audio feature reduction and analysis for automatic music genre classification, *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 5-8 Oct. 2014.

Berenzweig, A., Ellis, D. P. W. and Lawrence, S. (2002) Using Voice Segments to Improve Artist Classification of Music.

Bishop, C. M. (2006) *Pattern Recognition and Machine Learning*, *4*Springer.

Bogert, B. P., Healy, M. J. R. and Tukey, J. W. (1963) The Quefrency Alanysis of Time Series for Echoes: Cepstrum, Pseudo Autocovariance, Cross-Cepstrum and Saphe Cracking. *Proceedings of the Symposium on Time Series Analysis* Chapter 15, 209-243.

Breiman, L. (1996) Bagging Predictors. *Machine Learning*, 24(2), 123-140.

Breiman, L. (2001) Random Forests. *Machine Learning*, 45.

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984) *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group.

Brownlee, J. (2011) *Clever Algorithms: Nature-Inspired Programming Recipes*Lulu.com.

Burred, J. J. and Lerch, A. (2003) A Hierarchical Approach to Automatic Musical Genre Classification. *6th Int. conference on digital audio effects (DAFx-03), London, UK, September 8-11, 2003.*

Burred, J. J. and Lerch, A. (2004) Hierarchical automatic audio signal classification. *Journal of the Audio Engineering Society*, 52(7-8), 724-739.

Changseok, Bae, C., Chung, Y. Y., Shukran, M. A. M., Choi, E. and Yeh, W. C. (2008) An intelligent classif ication algorithm for LifeLog multimedia applications, *Multimedia Signal Processing, 2008 IEEE 10th Workshop on*. 8-10 Oct. 2008.

Chapra, S. C. and Canale, R. (2006) *Numerical Methods for Engineers*McGraw-Hill, Inc.

Chen, K., Wang, L. and Chi, H. (1997) Methods of Combining Multiple Classifiers with Different Features and Their Applications to Text-Independent Speaker Identification. *International Journal of Pattern Recognition and Artificial Intelligence*, 11(03), 417-445.

Chu, W. and Champagne, B. (2008) A Noise-Robust FFT-Based Auditory Spectrum With Application in Audio Classification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(1), 137-150.

Dadula, C. P. and Dadios, E. P. (2015) Neural network classification for detecting abnormal events in a public transport vehicle, *Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM), 2015 International Conference on*. 9-12 Dec. 2015.

Delgado-Contreras, J. R., Garcia-Vazquez, J. P. and Brena, R. F. (2014a) Classification of environmental audio signals using statistical time and frequency features, *Electronics, Communications and Computers (CONIELECOMP), 2014 International Conference on*. 26-28 Feb. 2014.

Delgado-Contreras, J. R., Garćıa-Vázquez, J. P., Brena, R. F., Galván-Tejada, C. E. and Galván-Tejada, J. I. (2014b) Feature Selection for Place Classification through Environmental Sounds. *Procedia Computer Science*, 37, 40-47.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977) Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B*, 39, 1-38.

Dennis, J. W. (2014) Sound Event Recognition in Unstructured Environments using Spectrogram Image Processing, Nanyang Technological University.

Dhanalakshmi, P., Palanivel, S. and Ramalingam, V. (2009) Classification of audio signals using SVM and RBFNN. *Expert Systems with Applications*, 36(3), 6069-6075.

Dhanalakshmi, P., Palanivel, S. and Ramalingam, V. (2010) Classification of audio signals using AANN and GMM. *Applied Soft Computing*.

Dogan, E., Sert, M. and Yazici, A. (2009) Content-Based Classification and Segmentation of Mixed-Type Audio by Using MPEG-7 Features, 152-157.

Ethem, A. (2014) Introduction to Machine LearningThe MIT Press.

Fiebrink, R. and Fujinaga, I. (2006) Feature Selection Pitfalls and Music Classification, ISMIR.

Gerhard, D. (2003) Audio Signal Classification: History and Current echniques (Technical Report TR-CS 2003-07).

Gharsalli, S., Emile, B., Laurent, H., Desquesnes, X. and Vivet, D. (2015) Random forest-based feature selection for emotion recognition, Image Processing Theory, Tools and Applications (IPTA), 2015 International Conference on. 10-13 Nov. 2015.

Giannakopoulos, T. and Pikrakis, A. (2014) Introduction to Audio Analysis: A MATLAB Approach. Elsevier Academic Press.

Gonzalez, R. C. and E., W. R. (2008) *Digital Image Processing (3rd Edition)*Prentice-Hall, Inc.

Grimaldi, M., Cunningham, P. and Kokaram, A. (2003) An Evaluation of Alternative Feature Selection Strategies and Ensemble Techniques for Classifying Music, *Workshop on Multimedia Discovery and Mining, 14th European Conference on Machine Learn-ing, 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*. Dubrovnik, Croatia, 102-108.

Guo, T., Wu, X., Yang, L., Huang, X. and Yu, M. (2014) Body part recognization base on hierarchy random forest with feature Pre-selection, *Signal Processing (ICSP), 2014 12th International Conference on*. 19-23 Oct. 2014.

Guyon, I. and Elisseeff, A. (2003) An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3, 1157-1182.

Hand, D. J., Adams, N. M. and Kelly, M. G. (2001) Multiple Classifier Systems Based on Interpretable Linear Classifiers, in Kittler, J. and Roli, F. (eds), *Multiple Classifier Systems*. Lecture Notes in Computer ScienceSpringer Berlin Heidelberg, 136-147.

Harrington, P. (2012) *Machine Learning in Action*Manning Publications Co.

Hastie, T., Tibshirani, R. and Friedman, J. (2016) *The Elements of Statistical Learning*Springer New York Inc.

Haykin, S. (1998) *Neural Networks: A Comprehensive Foundation*Prentice Hall PTR.

Hossain, M. S. and Muhammad, G. (2016) Healthcare Big Data Voice Pathology Assessment Framework. *IEEE Access*, 4, 7806-7815.

Hu, X. Q., Cui, M. and Chen, B. (2009) Feature selection based on random forest and application in correlation analysis of symptom and disease, *IT in Medicine and Education, 2009. ITIME '09. IEEE International Symposium on*. 14-16 Aug. 2009.

ISMIR (2015) *The 16th International Society for Music Information Retrieval Conference*, 2015. Available online: http://ismir2015.uma.es/ [Accessed.

Joelsson, S. R., Benediktsson, J. A. and Sveinsson, J. R. (2006) Feature Selection for Morphological Feature Extraction using Random Forests, *Signal Processing Symposium, 2006. NORSIG 2006. Proceedings of the 7th Nordic*. June 2006.

Jordan, M., Kleinberg, J. and Scholkopf, B. (2000) Independent component analysis: algorithms and applications. *Neural Netw.*, 13(4-5), 411-430.

Kayim, G., Sari, C. and Akgul, C. B. (2013) Facial feature selection for gender recognition based on random decision forests, *Signal Processing and Communications Applications Conference (SIU), 2013 21st*. 24-26 April 2013.

Kim, H. G., Moreau, N. and Sikora, T. (2004) Audio classification based on MPEG-7 spectral basis representations. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5), 716-725.

Kim, H. G., Moreau, N. and Sikora, T. (2005) MPEG-7 Audio and Beyond.

Kittler, J. and Devijver, P. A. (1982) Pattern Recognition, A Statistical Approach.

Kohavi, R. and John, G. H. (1997) Wrappers for feature subset selection. *Artificial Intelligence*, 97(1), 273-324.

Kos, M., Grasic, M., Vlaj, D. and Kacic, Z. (2009) On-Line Speech/Music Segmentation for Broadcast News Domain, *Systems, Signals and Image Processing, 2009. IWSSIP 2009. 16th International Conference on*. 18-20 June 2009.

Lampropoulos, A. S. and Tsihrintzis, G. A. (2012) Evaluation of MPEG-7 Descriptors for Speech Emotional Recognition, *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012 Eighth International Conference on*. 18-20 July 2012.

Lerch, A. (2012) Fundamentals, *An Introduction to Audio Content Analysis:Applications in Signal Processing and Music Informatics* Wiley-IEEE Press.

Liu, H. and Motoda, H. (1998) *Feature Selection for Knowledge Discovery and Data Mining*Kluwer Academic Publishers.

Liu, S. C., Bi, J., Jia, Z. Q., Chen, R., Chen, J. and Zhou, M. M. (2007) Automatic Audio Classification and Speaker Identification for Video Content Analysis, 91-96.

Mcadams, S. (1999) Perspectives on the Contribution of Timbre to Musical Structure. *Comput. Music J.*, 23(3), 85-102.

Mitrović, D., Zeppelzauer, M. and Breiteneder, C. (2010) Features for Content-Based Audio Retrieval, in Marvin, V. Z. (ed), *Advances in Computers*Elsevier, 71-150.

Molina, L. C., Belanche, L. and Nebot, A. (2002) Feature selection algorithms: a survey and experimental evaluation, *2002 IEEE International Conference on Data Mining, 2002. Proceedings.* 2002.

Moorer, J. (1974) The optimum comb method of pitch period analysis of continuous digitized speech. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 22(5), 330-338.

Muhammad, G. and Alghathbar, K. (2009) Environment Recognition from Audio Using MPEG-7 Features *2009 Fourth International Conference on Embedded and Multimedia Computing*. 10-12 Dec. 2009.

Murata, R., Mishina, Y., Yamauchi, Y., Yamashita, T. and Fujiyoshi, H. (2015) Efficient feature selection method using contribution ratio by random forest, *Frontiers of Computer Vision (FCV), 2015 21st Korea-Japan Joint Workshop on*. 28-30 Jan. 2015.

Murthy, Y. V. S. and Koolagud, i. G. (2015) Classification of vocal and non-vocal regions from audio songs using spectral features and pitch variations, *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*. 3-6 May 2015.

Neal, L., Briggs, F., Raich, R. and Fern, X. Z. (2011) Time-frequency segmentation of bird song in noisy acoustic environments, *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. 22-27 May 2011.

Paja, W. and Wrzesien, M. (2013) Melanoma important features selection using random forest approach, *Human System Interaction (HSI), 2013 The 6th International Conference on*. 6-8 June 2013.

Panagiotakis, C. and Tziritas, G. (2005) A speech/music discriminator based on RMS and zero-crossings. *Multimedia, IEEE Transactions on*, 7(1), 155-166.

Pang, H., George, S. L., Hui, K. and Tong, T. (2012) Gene Selection Using Iterative Feature Elimination Random Forests for Survival Outcomes. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 9(5), 1422-1431.

Peng, H., Long, F. and Ding, C. (2005) Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8), 1226-1238.

Phan, H., Hertel, L., M., M., Koch, P., Mazur, R. and Mertins, A. (2017) Improved Audio Scene Classification Based on Label-Tree Embeddings and Convolutional Neural Networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(6), 1278-1290.

Räsänen, O. and Pohjalainen, J. (2013) Random subset feature selection in automatic recognition of developmental disorders, affective states, and level of conflict from speech, *INTERSPEECH*.

Reif, D. M., Motsinger, A. A., McKinney, B. A., Crowe, J. E. and Moore, J. H. (2006) Feature Selection using a Random Forests Classifier for the Integrated Analysis of Multiple Data Types, *IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology, 2006. CIBCB*. 28-29 Sept. 2006.

Reynolds, D. A. and Rose, R. C. (1995) Robust text-independent speaker identification using Gaussian mixture speaker models. *Speech and Audio Processing, IEEE Transactions on*, 3(1), 72-83.

Rifkin, R. and A., K. (2004) In Defense of One-Vs-All Classification. *J. Mach. Learn. Res.*, 5, 101-141.

Rong, J., Li, G. and Chen, Y.-P. P. (2009) Acoustic feature selection for automatic emotion recognition from speech. *Information Processing and Management*, 45(3), 315-328.

Safavian, S. R. and Landgrebe, D. (1991) A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3), 660-674.

Salembier, P. and Sikora, T. (2002) *Introduction to MPEG-7: Multimedia Content Description Interface*John Wiley \andamp; Sons, Inc.

Segal, M. (2004). Machine learning benchmarks and random forest regression, Technical report, eScholarship Repository, University of California.

Sezgin, M. C., Günsel, B. and Kurt, G. K. (2011) A novel perceptual feature set for audio emotion recognition, *Face and Gesture 2011*. 21-25 March 2011.

Skurichina, M. (2001) *Stabilizing Weak Classifiers: Regularization and Combining Techniques in Discriminant Analysis*Technical University of Delft.

Song, Y., Wang, W. H. and Guo, F. J. (2009) Feature extraction and classification for audio information in news video, *Wavelet Analysis and Pattern Recognition, 2009. ICWAPR 2009. International Conference on*. 12-15 July 2009.

Sonnleitner, R., Niedermayer, B., Widmer, G. and Schlüter, J. (2012) A simple and effective spectral feature for speech detection in mixed audio signals, *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx'12)*.

Strang, G. (2009) Introduction to Linear Algebra.

Tang, J., Alelyani, S. and Liu, H. (2014) Feature Selection for Classification: A Review, *Data Classification*. Chapman and Hall/CRC Data Mining and Knowledge Discovery SeriesChapman and Hall/CRC, 37-64.

Temko, A. (2007) Acoustic Event Detection and Classification. Universitat Politècnica de Catalunya, Department of Signal Theory and Communications.

Thambi, S. V., Sreekumar, K. T., Kumar, C. S. and Raj, P. C. R. (2014) Random forest algorithm for improving the performance of speech/non-speech detection, *2014 First International Conference on Computational Systems and Communications (ICCSC)*. 17-18 Dec. 2014.

Theodoridis, S. and Koutroumbas, K. (2009) *Pattern Recognition, Fourth Edition*. Academic Press.

Tzanetakis, G. and Cook, F. (1999) A framework for audio analysis based on classification and temporal segmentation, *EUROMICRO Conference, 1999. Proceedings. 25th*. 1999.

Tzanetakis, G. and Cook, P. (2002) Musical genre classification of audio signals. *Speech and Audio Processing, IEEE Transactions on*, 10(5), 293-302.

Uddin, M. T. and Uddiny, M. A. (2015) A guided random forest based feature selection approach for activity recognition, *Electrical Engineering and Information Communication Technology (ICEEICT), 2015 International Conference on*. 21-23 May 2015.

Wang, L., Huang, S., Wang, S., Liang, J. and Xu, B. (2008) Music Genre Classification Based on Multiple Classifier Fusion, *Fourth International Conference on Natural Computation*. 18-20 Oct. 2008.

Wang, L. M., Chen, J. X., Fan, M., Zhao, X., Cui, H. T., Qou, M. J., Wang, S. X., Li, X. H., Jiang, Y. M., Zhou, L. Q. and Peng, X. (2010) Application of random forest data mining method to the feature selection for female sub-health state, *Bioinformatics and Biomedicine Workshops (BIBMW), 2010 IEEE International Conference on*. 18-18 Dec. 2010.

Webb, A. R. and Copsey, K. D. (2011) *Statistical Pattern Recognition*Wiley.

Webb, A. R., Lows, D. and Bedworth, M. D. (1988) A hybrid optimisation strategy for feed-forward adaptive layered networks. *DRA Memo 4193, DERA*.

Xiong, W. and Wang, C. (2009) A Hybrid Improved Ant Colony Optimization and Random Forests Feature Selection Method for Microarray Data, *INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference on*. 25-27 Aug. 2009.

Yamauchi, T. (2013) Mouse Trajectories and State Anxiety: Feature Selection with Random Forest, *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*. 2-5 Sept. 2013.

Yang, L. and Su, F. (2012) Auditory context classification using random forests, *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. 25-30 March 2012.

Zhang, J., Jiang, B., Lu, L. and Zhao, Q. (2010) Audio Segmentation System for Sport Games, *Electrical and Control Engineering (ICECE), 2010 International Conference on*. 25-27 June 2010.

Zhang, Y. F. and Lv, D. J. (2015) Selected Features for Classifying Environmental Audio Data with Random Forest. *The Open Automation and Control Systems Journal*, 7(1).

Ziyou, X., Radhakrishnan, R., Divakaran, A. and Huang, T. S. (2003) Comparing MFCC and MPEG-7 audio features for feature extraction, maximum likelihood HMM and entropic prior HMM for sports audio classification, *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*. 6-9 July 2003.

Appendix: Publications

- Al-Maathidi, M., and Li, F. (2012a). Feature Spaces And Machine Learning Regime For Audio Content Classification And Indexing. *SDIWC International Computer Science Conferences*, 262-274.
- Al-Maathidi, M., and Li, F. (2012b). NNet Based Audio Content Classification and Indexing System. [Jornal]. *International Journal of Digital Information and Wireless Communications (IJDIWC), 2*(4), 66-78.
- Al-Maathidi, M., and Li, F. (2014). Feature Spaces and Machine Learning Regimes for Audio Classification, A Comparative Study. *Proc. IEEE SIMS 2014*, 100-105.
- Mohammed, D. Y., Duncan, P. J., Al-Maathidi, M., and Li, F. F. (2015, 22-24 July 2015). *A system for semantic information extraction from mixed soundtracks deploying MARSYAS framework.* Paper presented at the Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on Industrial Informatics.
- Al-Maathidi, M., and Li, F. (2015, 18-20 December 2015). *Audio Content Feature Selection and Classification, A random Forests and Decision Tree Approach.* Paper presented at the International Conference on Progress in Informatics and Computing (PIC-2015), Nanjing, China.