AUDIO EVENT DETECTION ON TV BROADCAST

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EZGİ CAN OZAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2011

Approval of the thesis:

AUDIO EVENT DETECTION ON TV BROADCAST

submitted by **EZGİ CAN OZAN** in partial fulfilment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen _____
Head of Department, **Electrical and Electronics Engineering**

Assoc. Prof. Dr. Tolga Çiloğlu _____
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members

Prof. Dr. Mübeccel Demirekler _____
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. Tolga Çiloğlu _____
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. Çağatay Candan _____
Electrical and Electronics Engineering Dept., METU

Asst. Prof. Dr. Yeşim Serinağaoğlu Doğrusöz _____
Electrical and Electronics Engineering Dept., METU

M.Sc. Banu Oskay Acar _____
TÜBİTAK UZAY

Date: **15.09.2011**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Ezgi Can Ozan

Signature :

ABSTRACT

AUDIO EVENT DETECTION ON TV BROADCAST

Ozan, Ezgi Can

M.Sc., Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. Tolga Çiloğlu

February 2011, 65 pages

The availability of digital media has grown tremendously with the fast-paced ever-growing storage and communication technologies. As a result, today, we are facing a problem in indexing and browsing the huge amounts of multimedia data. This amount of data is impossible to be indexed or browsed by hand so automatic indexing and browsing systems are proposed. Audio Event Detection is a research area which tries to analyse the audio data in a semantic and perceptual manner, to bring a conceptual solution to this problem. In this thesis, a method for detecting several audio events in TV broadcast is proposed. The proposed method includes an audio segmentation stage to detect event boundaries. Broadcast audio is classified into 17 classes. The feature set for each event is obtained by using a feature selection algorithm to select suitable features among a large set of popular descriptors. Support Vector Machines and Gaussian Mixture Models are used as classifiers and the proposed system achieved an average recall rate of 88% for 17 different audio events. Comparing with the results in the literature, the proposed method is promising.

Keywords: Audio Event Detection, Audio Processing, Signal Processing, Audio Segmentation, Pattern Recognition.

ÖZ

TELEVİZYON YAYINLARINDA SES OLAY TESPİTİ

Ozan, Ezgi Can
Yüksek Lisans, Elektrik-Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Assoc. Prof. Dr. Tolga Çiloğlu

Eylül 2011, 65 sayfa

Gelişen teknolojiyle birlikte, görsel ve işitsel bilginin erişilebilirliği muazzam bir biçimde arttı. Bu artışın beraberinde getirdiği bir sonuç olarak, işitsel bilginin sınıflandırılması ve erişilmesi, bir sorun olarak karşımıza çıkmış bulunuyor. Bu büyüklükte bir verinin elle sınıflandırılması mümkün olmadığından, otomatik sistemler üzerinde çalışmalar devam ediyor. İşitsel Olay Sezimi, ses bilgisini algısal ve anlamsal olarak inceleyerek, bu probleme kavramsal bir yaklaşımla çözüm getirmeyi amaçlar. Bu çalışmada, televizyon yayınlarında yer alan ses olaylarının sezimi için bir yöntem önerilmiştir. Önerilen yöntem, ses olay sınırlarının tespiti için bir ses bölütleme metodu içerir. Televizyon yayını verisi 17 sınıf çerçevesinde incelenmiştir. Her ses sınıfı için uygun olan öznitelik seti, popular öznütelikler arasından bir öznitelik seçim algoritması kullanılarak seçilmiştir. Sınıflandırıcı olarak destek vektör makinaları ve Gauss karışım modelleri kullanılmış; ve önerilen metod 17 sınıf için ortalamada %88 lik bir doğruluk değeri elde etmiştir. Literatürdeki sonuçlarla karşılaştırıldığında, elde edilen sonuç ümit vericidir.

Anahtar Kelimeler: Ses Olay Sezimi, Ses Işleme, İşaret İşleme, Ses Bölütleme, Örüntü Tanıma.

To My Beloved Wife

ACKNOWLEDGMENTS

I would like to express my gratitude and deep appreciation to my supervisor Assoc. Prof. Dr. Tolga Çiloğlu for his guidance, positive suggestions and also for the great research environment he had provided.

I would like to also express my thanks for their assistance to Tugrul Kağan Ateş, İlkay Atıl, Cansu Gönülalan, Seda Tankız and Duygu Önür. With their help and support, preparation of this thesis became much easier.

I would like to thank my friends in Video and Audio Processing Group of Space Technologies Research Institute for the great research environment they had provided. I have learned so much from their experience and suggestions.

Finally, I would like to thank my wife, Gülcan, for her never ending love and support. This thesis is dedicated to her.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Content-based audio event detection is the problem of detecting predefined events on large databases of audio and it is a hot topic in the field of information retrieval, machine learning and pattern recognition. Audio databases may consist of isolated audio (as in sound effects databases) or mixed audio (as in broadcast) data. The event detection problem aims to find similar examples of a given audio event through a large set of audio data. Similarity definition depends on the aim of the problem.

Content-based detection of audio events is an important problem in today's world, considering the amount of present multimedia data. Huge databases of video and audio recordings are still growing, and so is the need for identifying the content and annotating these data for efficient classification and retrieval. Detection of audio events which are present in a multimedia data gives important information about the context and can be used in automatic annotation. Considering the amount of present data, manual annotation is neither possible nor feasible. Audio event detection research area aims to search through those databases and find out which audio events are there, giving important information about the context of data, and enabling annotation; and also retrieval.

Broadcast audio databases are different from the isolated audio databases in the sense that they consist of many different consecutive audio events, usually independent from each other. For isolated data, one can try to form a semantic or content-based relationship for the whole recording, as one movie or clip often contains audio data relevant with each other at some point, just like the shot concept in videos. In broadcast audio however, the main event can be interrupted many irrelevant audio events. Combined audio events are also frequently encountered on broadcast audio recordings. Therefore these databases are much more complex and harder to work on. For another example, detecting silence on broadcast audio recordings is a much more complex issue than detecting silence on isolated audio recordings, or manually segmented audio clips, since the audio event rapidly changes in broadcast recordings independently from each other. Determining a silence threshold is a

tough problem compared to working on isolated recordings since audio events change in isolated recordings much slower and the audio events present are often related to each other. For the aspect of content-based audio event detection problem, one can say that isolated audio databases are somewhat artificial, while broadcast audio is the real environment for this problem.

1.1. Scope

This study aims to propose a solution for the problem of detecting audio events on TV broadcasts. Audio event detection problem can be divided into sub-problems such as the problem of correctly determining the boundaries of audio events on TV broadcast, the problem of selecting suitable descriptors for each audio event, and the problem of selecting a suitable classifier. This study aims to bring a solution for these sub-problems.

The proposed solution starts with a segmentation phase, which is a preliminary step for detecting event boundaries, and which aims to handle the problem of boundary detection. A feature set is formed, which is prepared by collection of several audio descriptors that are most frequently used in the literature. The discriminating features for each audio event are determined by feature selection algorithms in order to find a solution for the problem of choosing suitable descriptors. Three different feature selection algorithms are tested in order to verify the feature selection performance. The selected descriptors are used to train classifiers for each event. Two different kinds of classifiers are tested for each event. The obtained results are compared with the given results in the literature.

Unlike the common case in the literature, which is classifying audio events using databases of isolated recordings, this study aims to detect audio events on TV broadcasts. Also the events in TV broadcasts are analyzed in detail. The common case in literature is describing audio events with three major classes; speech, music and others. This study aims to analyze the “other” class in detail, and detect different kinds of audio events in the “other” class separately.

1.2. Related Work

In this section, a review of the literature that has been investigated is presented. The literature is classified according to two main points of interest, namely, “audio segmentation” and “audio event detection”.

1.2.1. Audio Segmentation

Mainly, there are two approaches to the audio segmentation problem. One approach aims to divide the audio into *predefined* classes such as music, speech etc... Along this path, Lu et al. [1] make a speech–non-speech classification, and then non-speech regions are further classified into music and other events using a k-NN method.

Lu et al. [2] classify different type of audio events such as speech music and background sounds to compare classifiers like SVM, k-NN and GMM. They use features like MFCC, zero-crossing rate, short-time energy, brightness and band periodicity. SVM classifier gives the best classification accuracy which is near 90%.

Lu et al. [3] inspire from the concepts of video and text segmentation and proposed a new method describing key audio words. They described a semantic affinity measure, which determines the boundary point between two audio segments. This measure is directly proportional to frequency of occurring of audio elements in the segment, and inversely proportional to the time between audio elements. Segmenting the audio data, where semantic affinity is above a threshold value, they decide on the boundary locations. The proposed method is tested on data consisting of speech, music and applause sounds, and their different combinations. They describe their work as promising, which results a recall of nearly 70% for boundary locations.

Cai et al. [4] proposed an unsupervised method for dividing composite audio data into different events. Using spectral clustering, they divided the audio stream into classes like speech, music, noise, applause etc. These events are used to detect potential boundaries of audio segments. Then, they categorized these audio scenes in terms of audio events appearing in them.

The other approach to the audio segmentation is dividing the audio signal into segments using similarity measures which are used to detect the context change points. Yet the context of these segments is not predefined, and definition of this context is not part of the segmentation problem, as it is in the first approach.

Goodwin and Laroche [5] used a feature set including MFCC, zero-crossing rate and short-time energy; and measured the similarity of audio regions to divide the sound signal into segments. They used linear discriminant analysis method to improve the clustering of features, and applied dynamic programming to obtain better boundary points for the clusters.

S.Pfeifer [6] determined audio segments at different semantic levels using the method of pause and relative silence detection. Defining a minimum duration for pause and a maximum duration for interruption, Pfeifer showed that relevant pauses play an important role on detecting the segment boundaries.

J.Foote [7] proposed a method for automatically detecting the significant context changes in audio signals, using the self similarity of the signal, calculated by using kernel correlation method. Being an unsupervised method, Foote tested this algorithm on many application areas such as indexing and retrieval of audio data.

Tzanetakis and Cook [8] divide the audio signals into segments using Maholonobis distance metric that is calculated between consecutive frames aiming to detect the sudden changes of audio. They used features like spectral centroid, spectral roll-off, spectral flux and zero-crossing rate. They compared the results of the proposed algorithm with human reaction and show that, this reaction can be imitated using computer algorithms.

Zubari et al. [9] used energy based unsupervised segmentation method for detecting speech regions in the broadcast audio. They used short-time energy changes for segment boundary detection and they achieved a recall rate of 96% in speech regions.

Chen et al. [10] defined the audio segmentation problem as detecting the change points for speaker identity, environment or channel conditions. They modelled the audio signal as a Gaussian process and proposed using the Bayesian Information Criterion (BIC) to select the appropriate models for the given signal. BIC is used as a termination rule, so two segments are merged only if merging increases the BIC value. Chen et al. used the same BIC approach in [11] and Tritschler et al. used BIC in [12] and improved the accuracy and also the performance of the system.

Cettolo and Vescovi [13] used BIC for audio segmentation on radio news. They have tested different implementations of this algorithm and achieved 90% F1-score for detecting the boundaries.

Cheng et al. [14] proposed a “Divide-and-Conquer” algorithm for audio segmentation, based on the BIC. They developed a recursive algorithm that decreased the computational cost of BIC and also improved the accuracy.

1.2.2. Audio Event Detection

Growing amount of digital audio data makes it necessary to implement algorithms that automatically search and index the content of this data. Indexing and annotation by hand is neither possible nor efficient when the present yet increasing amount of the multimedia data is considered. There are many studies in the literature aiming to handle this problem, and some of them which we find interesting are reviewed.

Pfeiffer et al. [15] developed a framework using some basic features such as MFCC, spectrum centroid, spectrum flux, spectral roll-off, zero-crossing rate, fundamental frequency, harmonicity, etc. This framework is used in music indexing and violence detection. Recall rates of 81% for gunshot, 51% for cry and 93% for explosion sounds have been obtained on an isolated dataset. This can be regarded as one of the first studies in the field of audio event detection.

Zhang and Kuo [16] developed a system for audio event classification and retrieval, which classifies the audio into speech, music and environmental sounds. They used statistical and morphological properties of energy, fundamental frequency and zero crossing rate obtained by the audio signal. Using these features and their statistics, they first applied a model-free approach at the first stage of classification. This approach used relatively more basic features, which are easy to compute and extract. Whenever an abrupt change appears in any of these features, a boundary is set. Then they have also classified environmental sounds into sub-classes such as rain, bird and applause sounds using Gaussian Mixture Models. They used perceptual features like timbre and rhythm in this stage of classification. They used Hidden Markov Models (HMM) for model training and classification. As a result they achieved an accuracy rate of 80% on the average.

S. Li [17] presented a method for audio event classification and retrieval using nearest feature lines. Nearest feature lines method combines two samples of a given class by a linear equation and each point on this line is a sample for that class, creating infinite number of samples. The classification is done by computing the minimum distance between this feature line and the query sample.

Li considered most common perceptual and cepstral audio features and their combinations. He tested his method on a common audio database and the tests resulted with a nearly 10% error rate, which is significantly lower than those of the Nearest Neighbour (NN) based methods.

Li et al. [18] divided audio data into seven classes as single speaker speech, music, environmental noise, multiple speakers' speech, speech and music together, and speech and noise together. They tested 143 different features and the tests showed that cepstral features such as Mel-Frequency Cepstral Coefficients (MFCC) and linear prediction coefficients (LPC) brought better classification results compared to temporal and spectral features. They obtained an average 90% classification accuracy.

Guo and Li [19] used support vector machines (SVM) for discriminating 16 different audio event types. Using features like MFCC, spectrum power, brightness and pitch frequency, they compared the results of SVM with NN and k-NN and Nearest Center (NC) methods. The proposed method yields an average error rate of 11% for all 16 classes.

Wan and Lu [20] compared different features and distance measures. The features include MFCC, LPC, spectrum power and some other spectral/temporal and the distance measures include Euclidean distance, Kullback-Leibler (K-L) divergence, Mahalanobis distance and Bhattacharyya distance. As a result, K-L divergence and LPC are found to be the best distance measure and feature.

Baillie and Jose [21] developed an audio based event detection method which is used to detect special events in sports broadcasts. Using MFCC, they segmented and classified the audio stream at the same time using HMMs. Especially using the crowd reactions for specific events, this method resulted pretty well in soccer broadcasts for indexing and summarization.

Cai et al. [22] detected highlight sound effects in audio recordings using HMMs to model laughter, applause and crowd sounds. Using MFCC, energy, band power and zero crossing rates as features, they reported an average of 93% recall and 90% precision for those three audio events.

Portelo et al. [23] focused on detecting non-speech audio events, which consist of 15 different kinds of audio events such as jet sounds, bird sounds, vehicle sounds, telephone sounds, water sounds etc… They compared HMM and SVM classifiers and features like MFCC, PLP, spectrum power and pitch. Testing their method on movies and documentaries, Portela et al. demonstrated promising results, achieving a recall rate of 43% on the average.

Mesaros et al. [24] proposed a method for detecting acoustic events in real life recordings. They modeled each event by using MFCC features. They used HMMs as classifiers while segmenting the audio data at the same time. They reached an accuracy of 23% while classifying the real life audio recordings among 61 classes.

Temko et al. [25] developed a method for audio event detection and tested their method in the CLEAR 2007. CLEAR is an evaluation workshop, which is supported by NIST. CLEAR database consists of audio events such as door knocking, step sounds, spoon clings, paper wrapping, applause and telephone sounds other than speech, music and silence. The method used features like frequency-filtered band energies and perceptual features. Using SVMs as classifier, they detected events like door knock, keyboard typing, laughter, steps etc. They gained 30% recall and 20% precision at the end of the tests.

1.3. Outline

This thesis contains six chapters. Chapter 1 is the introduction chapter, in which the scope of this study is presented and previous studies in the literature are reviewed. In Chapter 2 the segmentation algorithm and the test results which verify the proposed hypothesises are presented. In Chapter 3, audio features that are used in the audio event detection method are described. The feature selection algorithms and classifiers are also presented in Chapter 3. Chapter 4 presents the experiments and the results for the audio event detection method. Comments about the obtained results are also presented. Finally in Chapter 5, the study is summarized and possible future works are presented.

CHAPTER 2

AUDIO SEGMENTATION

Audio segmentation is a preliminary step for audio event detection and retrieval on broadcast audio streams. In audio segmentation, it is desired to locate the boundaries on the audio stream where a single audio event is present in between. For some problems, these boundaries are speaker change points [26,27], for some problems they are phone change points [11] and for some problems like audio event detection, these boundaries are the audio event change points [28,29]. Audio segmentation aims to divide the audio signal into smaller regions, transforming the problem of detecting audio events on broadcast data to the problem of detecting audio events on an isolated dataset. The main purpose of an audio segmentation algorithm is to obtain small pieces of audio consisting of a single audio event and to determine the boundaries of event changes with high precision. The desired audio segmentation algorithm should not yield more than one audio event in a single segment, and should not divide a single audio event unnecessarily.

Finding the event change boundaries on a continuous broadcast stream is still a common and actual problem in audio pattern recognition area. There are many approaches to this problem in the literature [3,5,8-14,27,30,31]. One of the most commonly used methods is Bayesian Information Criterion. There are several approaches in the implementation of this method [10-14]. The most common approach is the divide and conquer algorithm in [14]. The details of this algorithm are given in Section 2.2. In this study, BIC algorithm is compared with Unsupervised Energy-Based Segmentation [9], and a combination of two algorithms is proposed. The proposed algorithm uses energy based segmentation to locate the segment boundary locations and BIC method to merge over-divided segments and increase the precision of segmentation.

2.1. Unsupervised Energy Segmentation Method

Unsupervised Energy Segmentation (UES) is a method for audio segmentation using short-time energy of consecutive audio frames [9]. UES method aims to detect segment boundary points, dividing the audio stream into "homogenous" regions.

2.1.1. Definition of Homogeneity

A homogenous region can be defined as a small part of audio, in which the type of the audio event does not change. The audio event present at the beginning of a homogenous segment is also the event at the end. For example, a segment is homogenous if it consists of speech only. Likewise, if a segment consists of speech and music together, it is a homogenous segment if the speech and music events are present within the segment from the beginning until the end of the segment, continuously and simultaneously. Detecting these homogeneous regions assures that, every short-time feature, extracted between these boundary points, belongs to the same audio event class. Whether these audio classes consist of one or more types of audio, we have the general assumption that every feature extracted from the frames in that segment are "close" to each other, provided that the right type of feature is selected.

2.1.2. Short-Time Energy

Short-time energy (STE) is the main feature used in UES. STE is the energy of an audio frame. STE is computed from frames of 10ms length and no overlap as

$$Energy = \sum_{x=0}^{N} x^2 \qquad 1$$

2.1.3. Detection of Segment Boundaries

The pseudo-code of the segment boundary detection algorithm is given in Figure 1. Firstly, two consecutive windows of 20-frames each are constructed. The powers of these windows are calculated. Then the greater power is divided by the lesser power to obtain a power ratio value. The windows slide one frame and the power ratios are computed at every step, generating a sequence. The local maxima points of this sequence, which are above a certain threshold value, are selected as segment boundaries.

```
Begin
// EnergySequence: Energy frame sequence for audio data.
// RatioSequence: Sequence of calculated power ratios

      EnergySequence: {f1, f2 …., fN}
      for each fi in EnergySequence
      {
            W1:{fi … fi+19}
            W2:{fi+20 … fi+39}
            PowerRatio = ∑W1 / ∑W2
            if (PowerRatio<1)
                  PowerRatio = 1 / PowerRatio

            if (PowerRatio>PThreshold)
                  RatioSequence ← PowerRatio
            else
                  RatioSequence ← 0
      }

      RatioSequence: {rs1, rs2,…rsN}
      for each rsi in RatioSequence
      {
            if(rsi > rsi+1 and rsi > rsi-1)
                  SegmentBoundaries ← i
      }
End
```

Figure 1: Segment Boundary Detection Pseudo-Code

2.1.4. Parameter Selection

Energy based segmentation algorithm has two parameters to adjust. One of them is the window length, which determines how many energy frames are used to calculate the window power. A window with more frames is more robust to sudden energy changes, which decreases the frequency of over-segmentation. But window with a greater size is less likely to detect segments with small durations, especially segments shorter than the window size, which leads to missing segment boundaries.

The second parameter of UES is the threshold which determines a local maximum point on the power ratio sequence is a segment boundary or not. Higher threshold values requires

sharper power changes on the signal, which may lead to under-segmentation in some cases, and lower threshold values are more sensitive to less sharp power changes, leading to over-segmentation.

To determine those two parameters, two performance measures are proposed. The first one is the "*success rate*" (SR) which evaluates the success of the localization of the determined segment boundaries. SR is defined as the total duration of segments which are found to be homogeneous divided by the total duration of the test set. The expression is given by Equation 2.

$$SR = \frac{\sum D_{Homogenous}}{\sum D} \quad (2)$$

D: Duration of segments

$D_{Homogenous}$: Duration of homogenous segments

The second performance measure is the "*average duration*" (AD) of the segments. The average duration is given by Equation 3.

$$AD = \frac{Total\ Duration\ of\ Segments}{Total\ Number\ of\ Segments} \quad (3)$$

Based on these two performance measures, a cost function is proposed to be minimized to select the parameters giving the best segmentation results, given in Equation 4.

$$Cost = 100 \times (1 - SR) - 4 \times AD + 1 \quad (4)$$

Using this function, tests have been performed to obtain the parameter values. The error rate, (1-SR), is given 25 times more weight than the average duration (AD) value. 1 is added to make sure the cost value is positive, for visualization purposes only. The change of cost function, success rate and average segment duration with respect to each other is given in Figure 2-7.

The tests have been performed on a 1 hour dataset annotated manually. The test data have been collected from 6 different TV channels, each having duration of 10 minutes. The broadcast types of channels have been selected different from each other in order to represent general TV broadcasting. Program types like commercials, movies, series, news and various are all represented in the collected data.

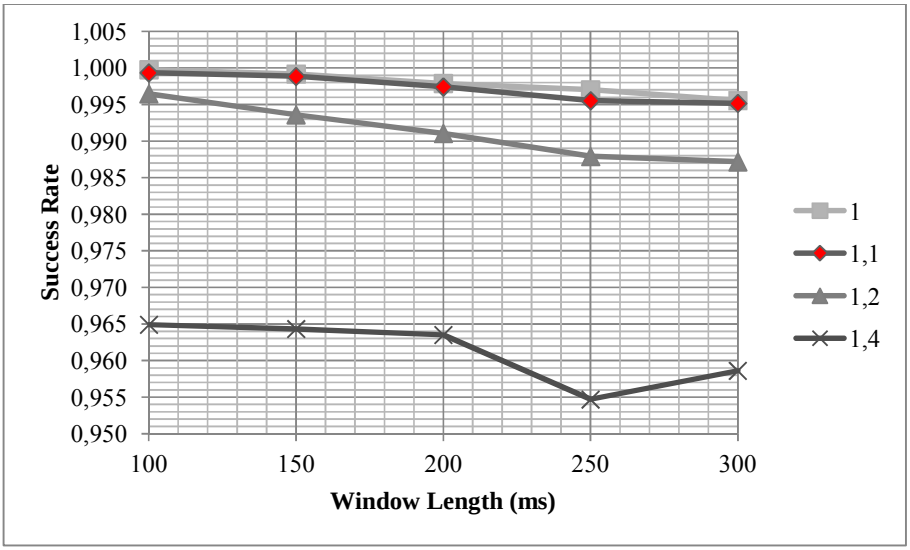


Figure 2: SR vs Window Length (vs Threshold)

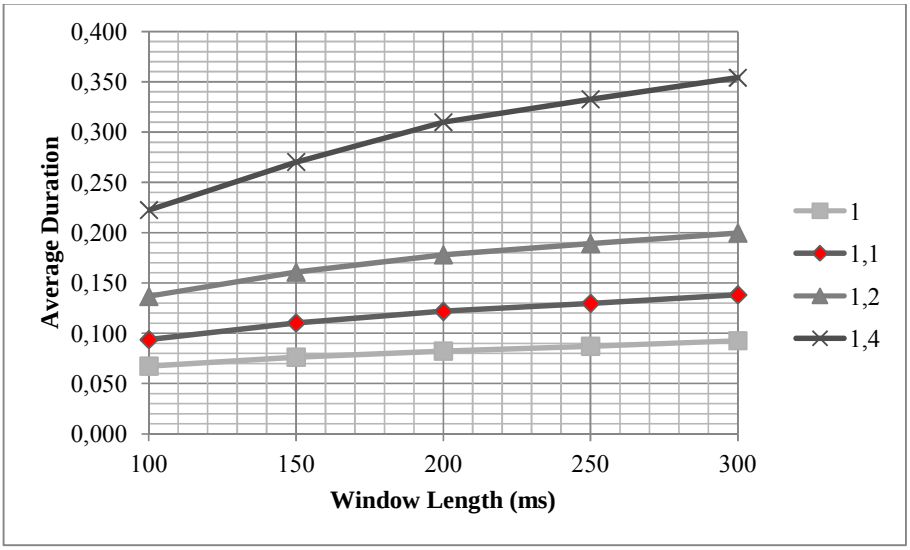


Figure 3 : AD vs Window Length (vs Threshold)

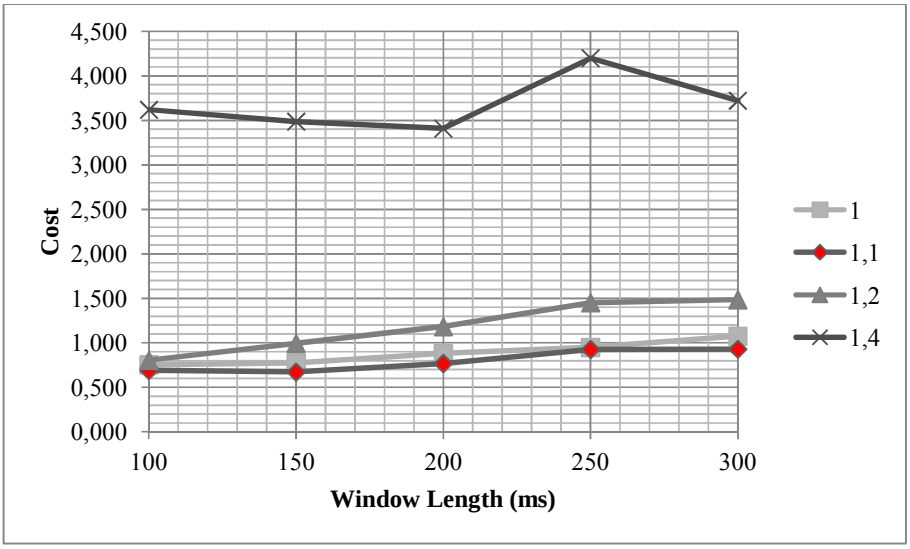


Figure 4: Cost vs Window Length (vs Threshold)

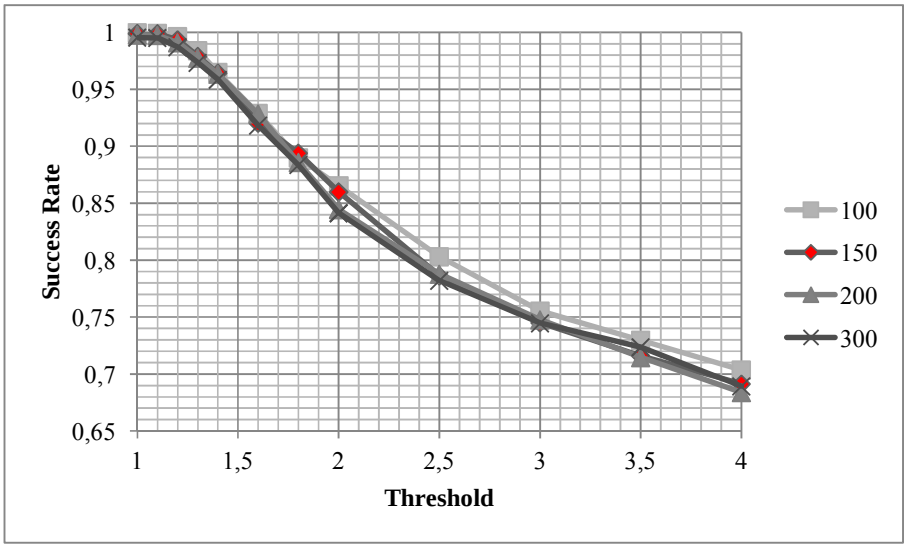


Figure 5: SR vs Threshold (vs Window Length)

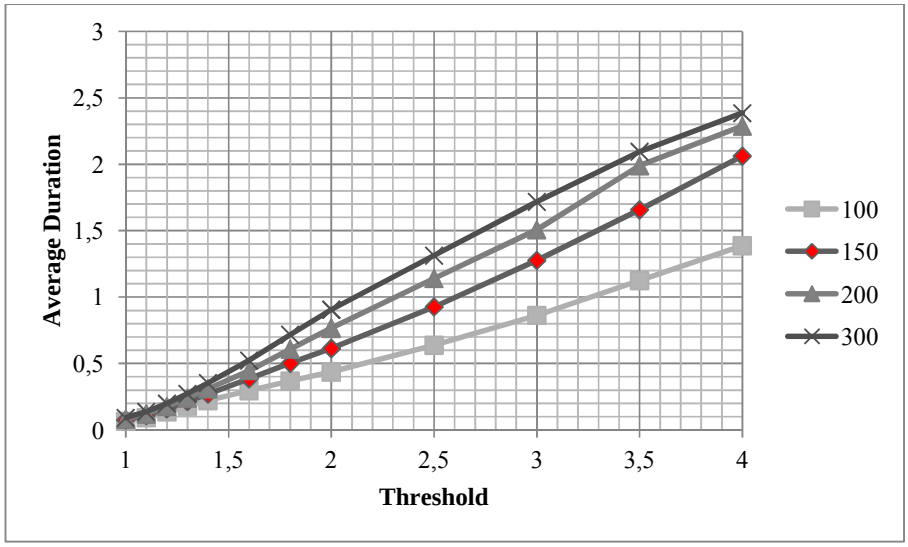


Figure 6: AD vs Threshold (vs Window Length)

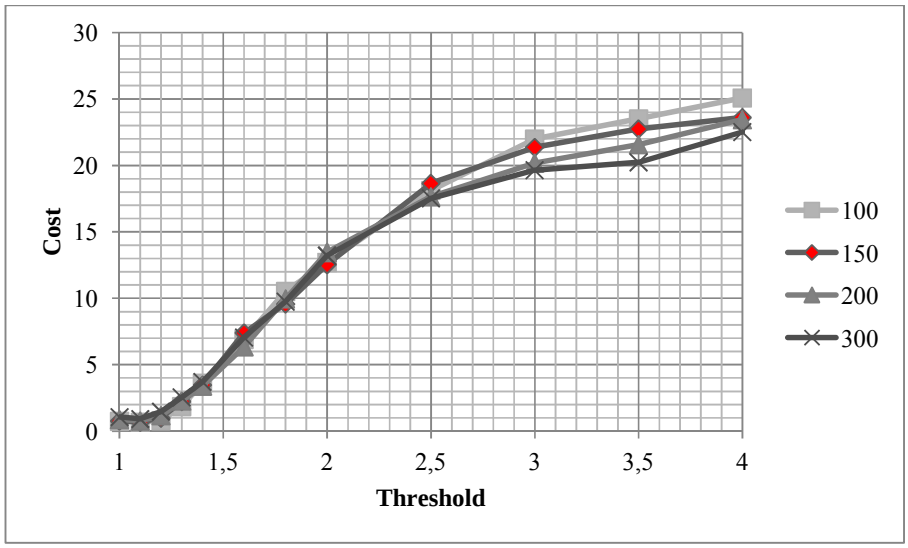


Figure 7: Cost vs Threshold (vs Window Length)

The tests performed on the given data set show that, the increase in the window length increases the average duration, but decreases the success rate faster, causing an increase in the cost function. Likewise, increase in the threshold value also increases the average duration but decreases the success rate faster also in this case, causing a steep increase in the cost value. According to the predefined cost function, an optimum value is selected. Best window length is selected to be 150ms and best threshold value is selected to be 1.10.

2.2. Bayesian Information Criterion Based Segmentation

The main idea in BIC based segmentation is to divide the data into segments considering a cost function using BIC [14]. This approach aims to select the best model for a given dataset. For every model M_i, and dataset Z, the BIC cost function is given in Equation 5.

$$\mathrm{BIC}(M_i, Z) = \log p(Z \mid \Theta_i) - 1/2\, \lambda\, \#(M_i)\, \log n \tag{5}$$

Here, Θ_i is the maximum likelihood estimate of parameters of M_i and $\#(M_i)$ is the number of parameters of M_i (in the given case, number of mixtures of the model) n is the dimension of the feature vector and λ is the penalty factor. The model corresponding to the highest BIC value is the most suitable model for the given dataset.

In BIC based segmentation, for every change point candidate, the hypothesis given in Equation 6 is tested with the function given in Equation 7.

$$\begin{aligned} &H0: z_1, z_2, \cdots, z_n \sim N(\mu, \Sigma) \\ &H1: z_1, z_2, \cdots, z_i \sim N(\mu_1, \Sigma_1) \text{ and } z_{i+1}, z_{i+2}, \ldots, z_n \sim N(\mu_2, \Sigma_2) \end{aligned} \tag{6}$$

$$\Delta BIC(i) = BIC(H_1, Z) - BIC(H_0, Z), i = 1, \cdots, n \tag{7}$$

In Equation 6, Hypothesis 0 assumes that, all the given samples, from z_1 to z_n belong to a single model. On the contrary, Hypothesis 1 states that given samples belong to two separate models. Samples from z_1 to z_i belong to model with the distribution $N(\mu_1, \Sigma_1)$ and samples from z_{i+1} to z_n belong to model with distribution $N(\mu_2, \Sigma_2)$. Therefore $\Delta BIC(i)$ is the difference between two BIC values computed with these two different assumptions.

If $\Delta BIC(i) > 0$, then i is selected to be a change point. Here $BIC(H_0, Z)$ is the BIC cost of modeling the dataset as a single model, and $BIC(H_1, Z)$ is the BIC cost which is calculated at a segment change point in i. So if modeling the given dataset in a single model is more costly then dividing into two segments, the point i is selected as a change point. In Equation 5, the term λ plays an important role on determining the cost of segmentation of the dataset.

The BIC based segmentation algorithm used in this study is based on the algorithm given in [14]. Accordingly, a sliding window approach is followed to detect the segment boundaries on the audio stream. This window has an initial length N_{ini} and increased by N_g until a segment change point is found or the window length reaches a termination value of N_{max}. If a change point is found, the window is reset to N_{ini} value; and the algorithm goes on from the last found change point. If no change point is found, the starting index is increased by N_s and the same operations above are applied.

Tests on BIC based segmentation method have been performed on the same dataset with UES. Three different kinds of audio features are used to represent the audio data. The features used are Mel-Frequency Cepstral Coefficients (MFCC), Spectrum Band Power (SBP) and Perceptive Linear Prediction Coefficients (PLP) respectively. The detailed descriptions of these features are given in Section 3. These features are selected considering their ability to model the spectral behavior of audio signals. These features are also widely used in the literature on segmentation problems [2,8,14,18,20,23,30,32-34]. Obtained test results are given in Figure 8 to Figure 10. The λ value, which determines the cost of segmentation in a given dataset, is also tested for different values and the best λ for each feature value is determined. The cost function in Equation 8 is used to select the best parameter for this segmentation method.

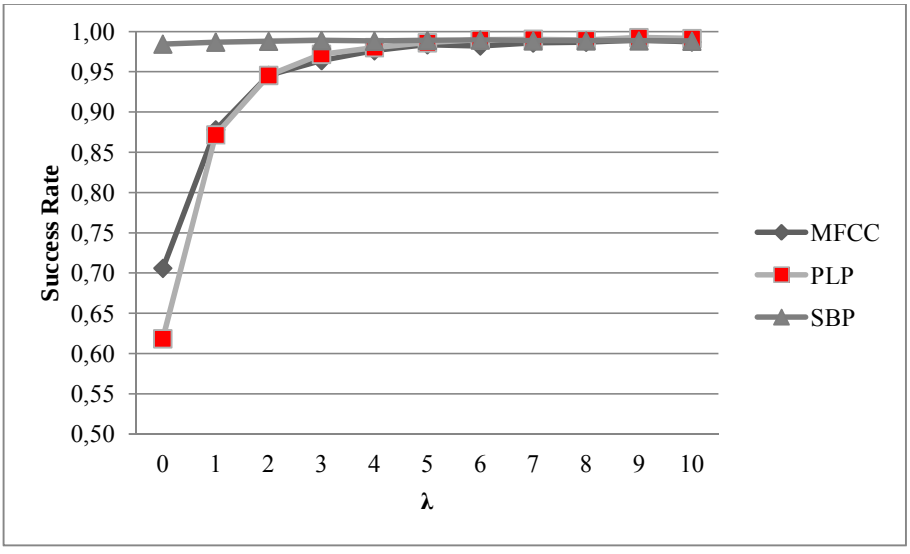


Figure 8: Success Rate vs λ

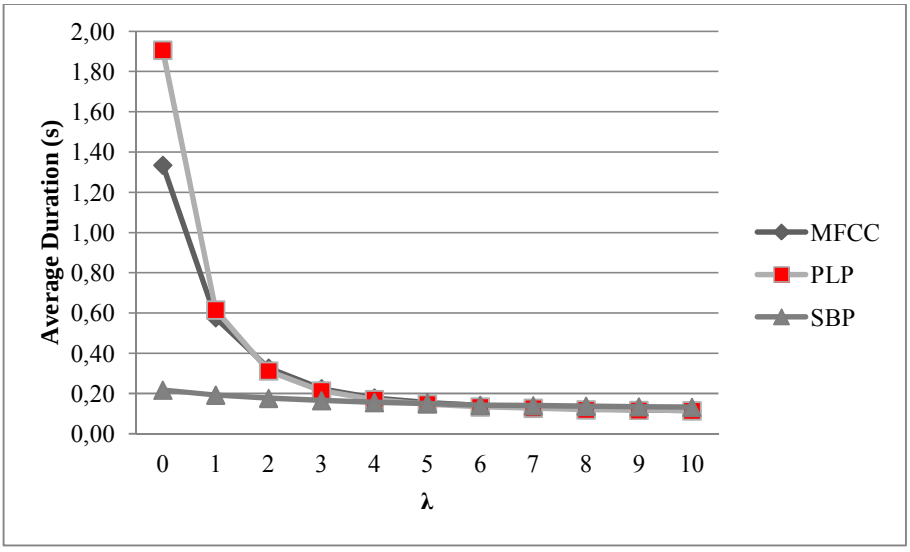


Figure 9: Average Duration vs λ

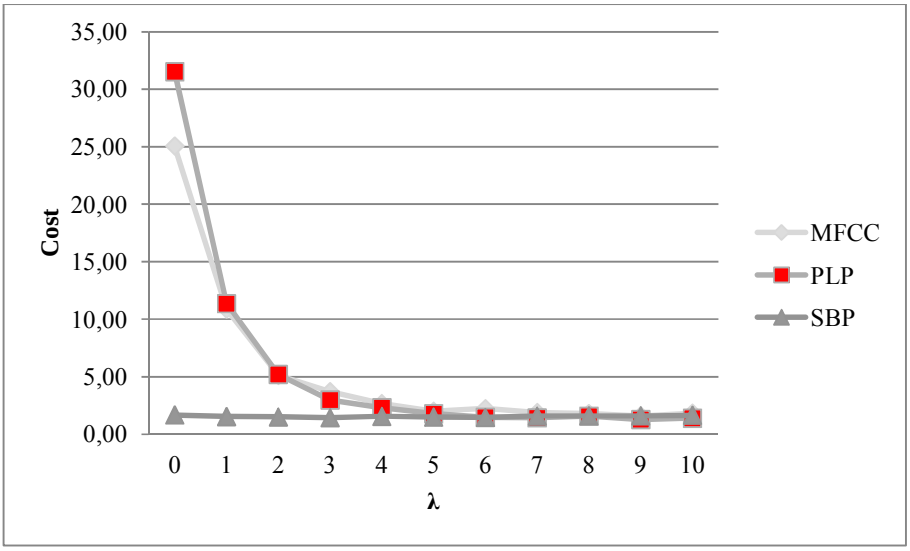


Figure 10: Cost vs λ

According to the performed tests, best feature for BIC segmentation is selected to be PLP with the λ value of 9. Increasing the value of λ and favoring the segmentation, decreases the average duration; yet increasing the success rate, causes a drop in the cost value.

2.3. Combination of BIC and UES

Considering the tests performed, one can state that UES algorithm is successful in detecting the segment boundary locations. Considering these locations as boundary candidates before

BIC is applied, a hybrid method can be proposed. This method has the advantages of both algorithms. Using UES as the boundary location selector, the major changes in the energy levels have been detected and segment homogeneity is satisfied with a fast algorithm. Combining these segments with BIC method brings the spectral information into segmentation problem, yielding a result of greater average duration values. Also predefined segments, located by UES algorithm, provides an isolated dataset for the BIC algorithm to decide on segmentation, to increase the success rate of BIC algorithm.

The proposed algorithm, which consists of the combination of UES and BIC methods, is given as a pseudo-code in Figure 11. In the proposed algorithm, first UES algorithm is run over the audio data, to determine the segment location candidates. Then two consecutive segments are given to BIC algorithm as one whole dataset, checking the boundary between them using the BIC cost function. If the calculated BIC cost is greater than zero, that point is selected as a segment boundary. Else, the segments are merged, and the algorithm goes on, accepting these two segments as one and computing the BIC cost value with the next segment.

```
Begin
      // EnergySegments: Segment candidates obtained by energy
      segmentation method.
      // BICSegments: Final segments after BIC merging.

      EnergySegments : {s0,s1….sn}
      for each si in EnergySegments
            if( BIC(si,si+1) ≤ 0 )
                  si ← si ∪ si+1
                  i = i - 1
            else
                  BICSegments ← si
End
```

Figure 11: UES and BIC Combination Pseudo-Code

Tests have been performed on this method to compare with the two previously defined methods. Same cost functions are used to compare the results obtained from these tests. The comparative results obtained from the tests are given in Figure 12 to Figure 14. The best parameters obtained in both UES-only and BIC-only methods have been used in the combination of those two methods. The window size of UES is selected to be 150ms; the

threshold is selected as 1.1. The feature for BIC method is selected to be PLP and the corresponding λ value is 9.

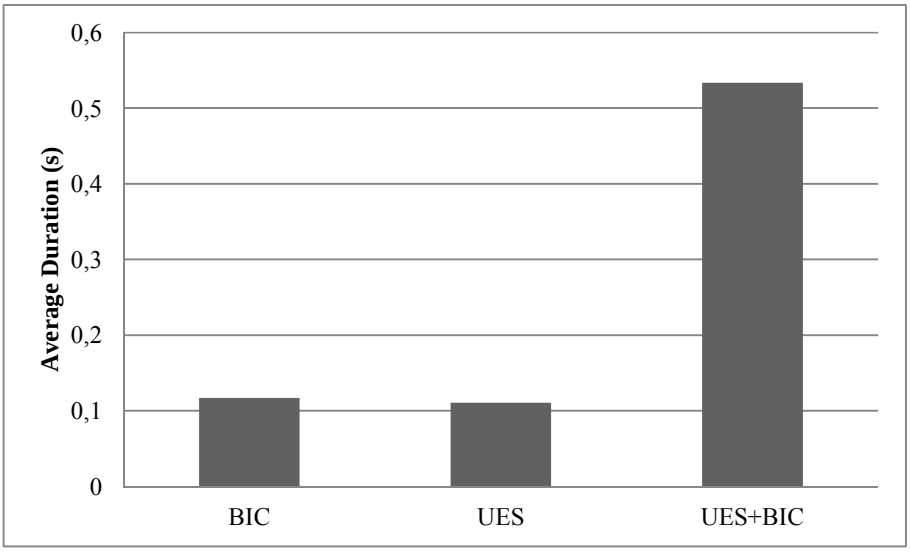


Figure 12: Average Duration of Methods

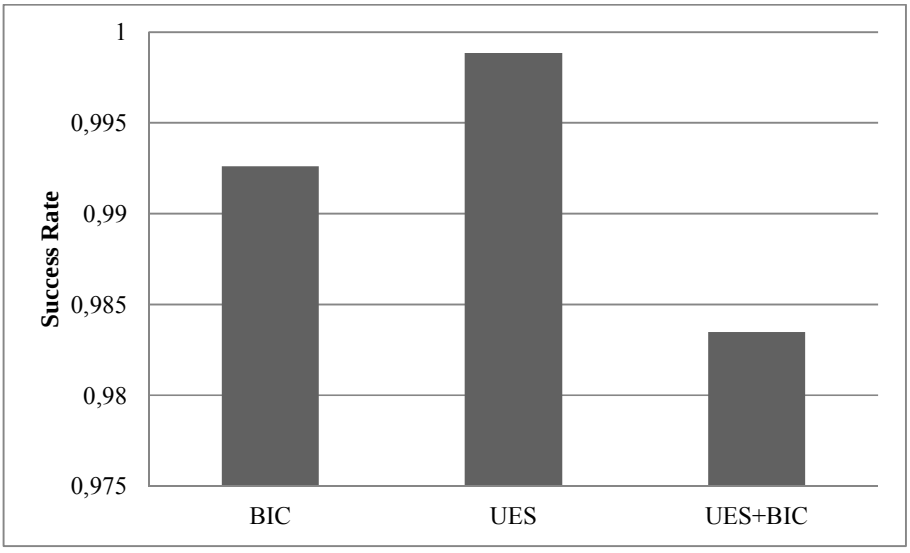


Figure 13: Success Rates of Methods

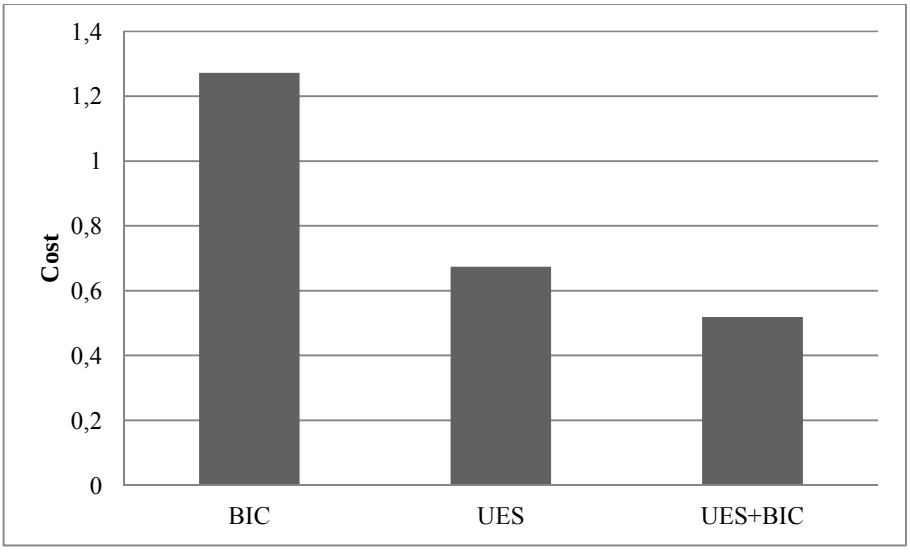


Figure 14: Costs of Methods

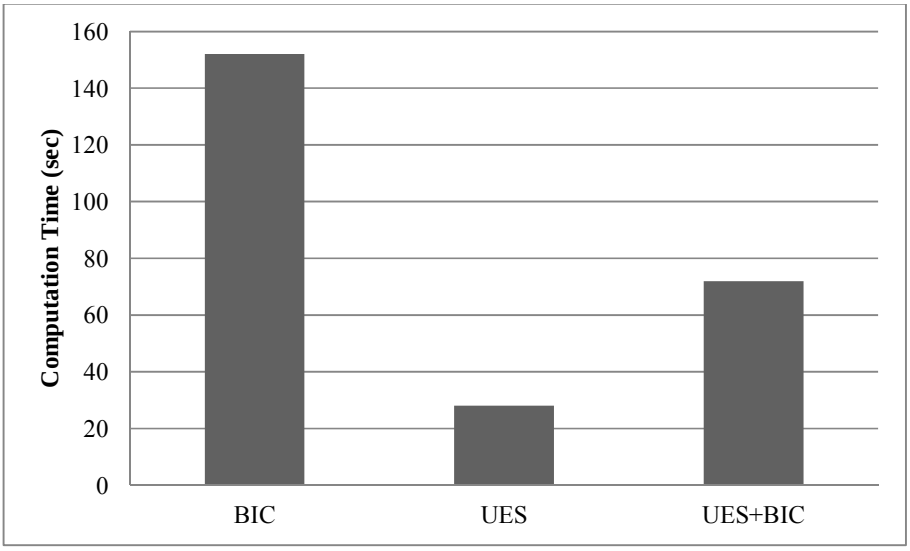


Figure 15: Computation Times of Methods

The tests prove that combining the two methods increases the average segment duration without decreasing the success rates so that it brings an improvement on the cost function. Computation time of BIC algorithm is also improved.

2.4. Activity/Non-Activity Region Detection

2.4.1. Definition of Activity / Non-Activity Regions

Audio data can be divided into two different kinds of regions as *Activity* and *Non-Activity*. Non-Activity regions can be defined as the regions that do not carry any significant aural information. Most of the non-activity regions are silence. Other than silence, background noise can be accepted as non-activity region. Since the context of noisy background areas are not easy to discriminate, and since those areas consist of very different sorts of sounds, elimination of these areas is expected to increase the precision of any audio event detection system. The background noises between the words of a conversation that is recorded outdoors, or the background music in a movie scene can also be accepted as non-activity regions. Any other region that is carrying significant aural information belonging to any audio event is called an activity region.

There are many different silence/background noise detection methods in the literature [1,2,6,9,16,18,27,30,35]. Most of them rely on a predefined or trained threshold value, or adaptive threshold values computed from the audio stream that is being examined. Because of the fast changing and dynamic nature of broadcast audio, it is usually not possible to determine a unique threshold, which can be accepted as a universal value. Even in the same audio recording, silence and background noise energy thresholds may vary because of the changing scene and broadcasting types. An adaptive method is selected to determine the activity and non-activity regions on the broadcast audio [9]. In this method, activity and non-activity transitions are modelled by using normalized power values of segments and according to this model, an adaptive decision mechanism is constructed to give activity and non-activity decision for segments.

2.4.2. Detection of Activity/Non-Activity Regions

Silence is defined as the lack of audible sound [36], and it is a relative definition because of the term “audible”. The audibility of a sound is relative to the general energy level of a recording. If a recording consists of generally high energy regions, the audibility of relatively lower energy regions can be in question. On the contrary, if a recording consists of generally low energy regions, this time the discrimination between audible and inaudible is harder to make. These are all because of the relativity of human nature and adaptability of human ear to different kinds of environments. In loud and noisy environments, human ear focuses on the loud sounds, which makes relatively less loud sounds inaudible. In silent,

quiet environments, human ear is adapted to hear sounds with less energy values, so former inaudible sounds are now audible. The definition and decision of a sound by the means of audibility is only meaningful by observing the whole environment, in the case of this study, considering the whole recording.

The inconsistency of energy levels of different broadcast recordings forces one to use an adaptive algorithm, which uses the energy information of consecutive segments. Since the energy levels may change between different recordings and different scenes, a silence threshold value which is universally applicable is not possible to find. The definition of silence is only meaningful with non-silent areas around, one should define both silence and non-silence, in other words, activity and non-activity regions together. Using this idea, a detection algorithm is proposed.

The flowchart of the Activity / Non-activity detection algorithm is given in Figure 16. First audio data is segmented into homogeneous segments using the segmentation algorithm. Then power of each segment is calculated. These power values are concatenated to obtain a feature vector of five dimensions. For each segment, the power of that segment, powers of two successors and two predecessors are concatenated, and Gaussian Mixture Models of activity and non-activity regions are trained using these feature vectors. Activity / Non-Activity models obtained are then used to decide whether a segment belongs to an activity or non-activity region.

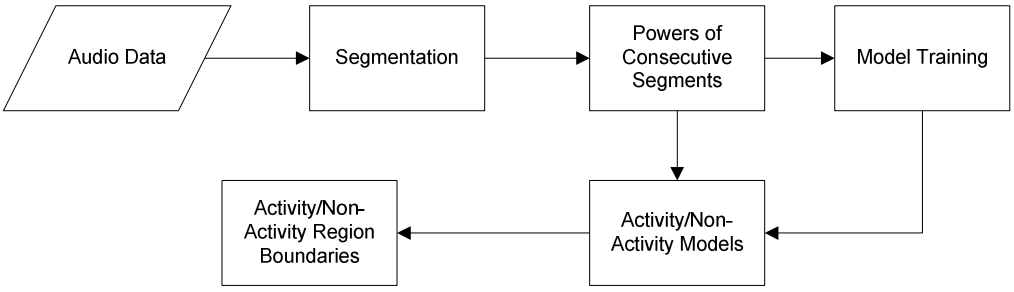


Figure 16: Activity / Non-Activity Detection Algorithm

The tests and obtained results of Activity/Non-Activity region detection are given in Figure 17-11. Different mixture numbers and different feature vector dimensions have been tested. The number of mixtures is tested in the range 2-256. Three different dimension values have been tested; 2, 3 and 5. 2 dimensional vector, consisting of the powers of each segment itself and its successor; 3 dimensional vector, consisting of the powers of each segment

itself, its one successor and one predecessor; and 5 dimensional vector consisting of the powers of each segment itself, its two successors and two predecessors. The test results show that best parameters for Activity/Non-Activity Region detection are 5 dimensional vectors trained with a GMM of 64 mixtures. The best parameters are selected according to the cost function given by Equation 8. The values minimizing the cost function are selected as optimum parameters.

$$R_A = \frac{S_{Relevant} \cap S_{Retrieved}}{S_{Relevant}} \qquad 8$$

$$P_A = \frac{S_{Relevant} \cap S_{Retrieved}}{S_{Retrieved}}$$

$$Cost = \frac{1}{R_A \times P_A}$$

R_A : Recall of Activity Regions

P_A : Precision of Activity Regions

The cost function in Equation 8 is determined considering the expected performance of the segmentation method. Precision and recall of the given system is calculated using the formula given in Equation 8, and the better precision and recall, the better the performance. So multiplication of those two values is used. To achieve the best performance, where the cost function is at minimum, the inverse of the multiplication of precision and recall is selected as the cost function.

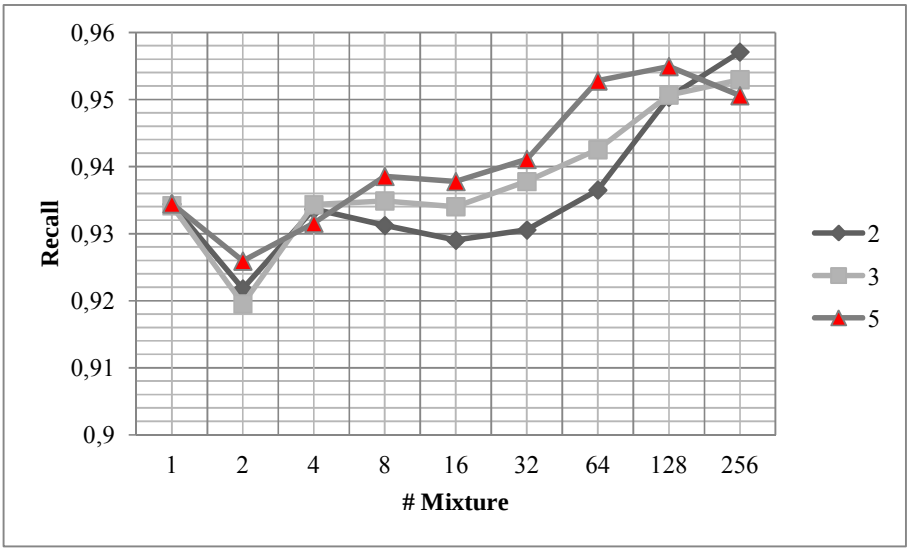


Figure 17: Recall vs #Mixture (vs Feature Dimension)

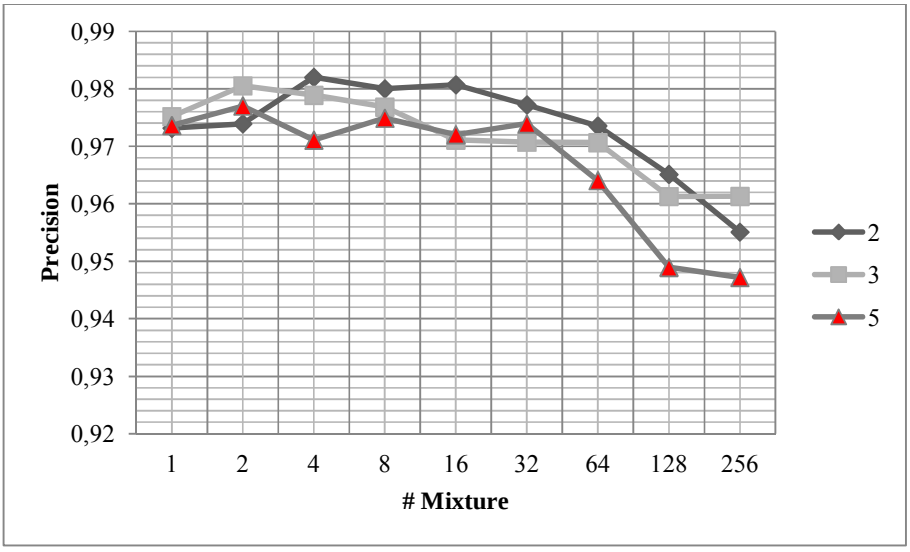


Figure 18: Precision vs #Mixture (vs Feature Dimension)

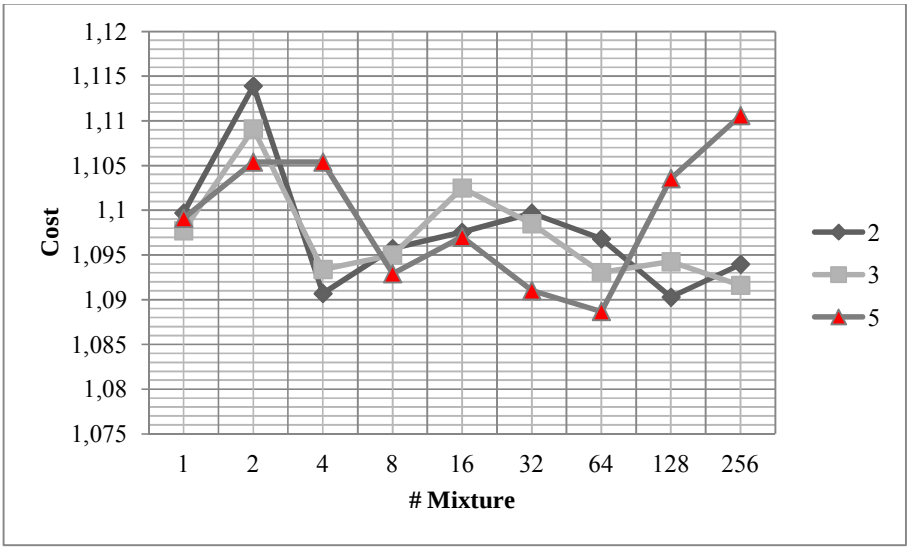


Figure 19: Cost vs #Mixture (vs Feature Dimension)

2.4.3. Normalization with Equivalent Power Regions

The problem of inconsistency of energy levels for different scenes in a broadcast recording is dealt with UES. If UES can deal with the inconsistency of energy levels of different recordings, the performance of Activity/Non-Activity detection can be improved and different scenes are also detected and treated as different recordings. Detection of

equivalent power regions aims to improve the performance of Activity/Non-Activity detection algorithm.

Definition of Equivalent Power Regions

An *Equivalent Power Region* (EPR) can be defined as a part of an audio recording, in which the recording energy level can be considered as constant. The boundaries of these EPR regions are detected using the algorithm in [9].

Detection of Equivalent Power Regions

EPR boundaries are detected using an algorithm described in Figure 20. In this algorithm audio data is divided into homogeneous segments using UES, then activity/non-activity region detection algorithm is applied, assuming that the whole recording consists of a single EPR. Then an algorithm which is similar to the UES is applied. Two sliding windows consisting of "a number of non-activity segments" is formed and traversed among the whole recording. The reason to use only the non-activity segments while calculating the powers of sliding windows is the difference of the variances of the powers of different types of segments in an EPR. The variance of the powers of non-activity segments in an EPR is observed to be much less than the variance of the powers of activity segments, so powers of non-activity segments is selected as a discriminating property between two consecutive EPRs [9]. The powers inside these windows are calculated and again as in UES algorithm, the greater power is divided by the lesser power, obtaining a sequence of power ratios. Detection of local maximum points on this sequence above a selected threshold value gives the boundaries of EPR regions. Then the powers of each segment inside an EPR are recalculated and normalized by the average power value of the corresponding EPR. Obtained power values are used to recreate features of Activity/Non-Activity detection algorithm and Activity/Non-Activity region boundaries are corrected.

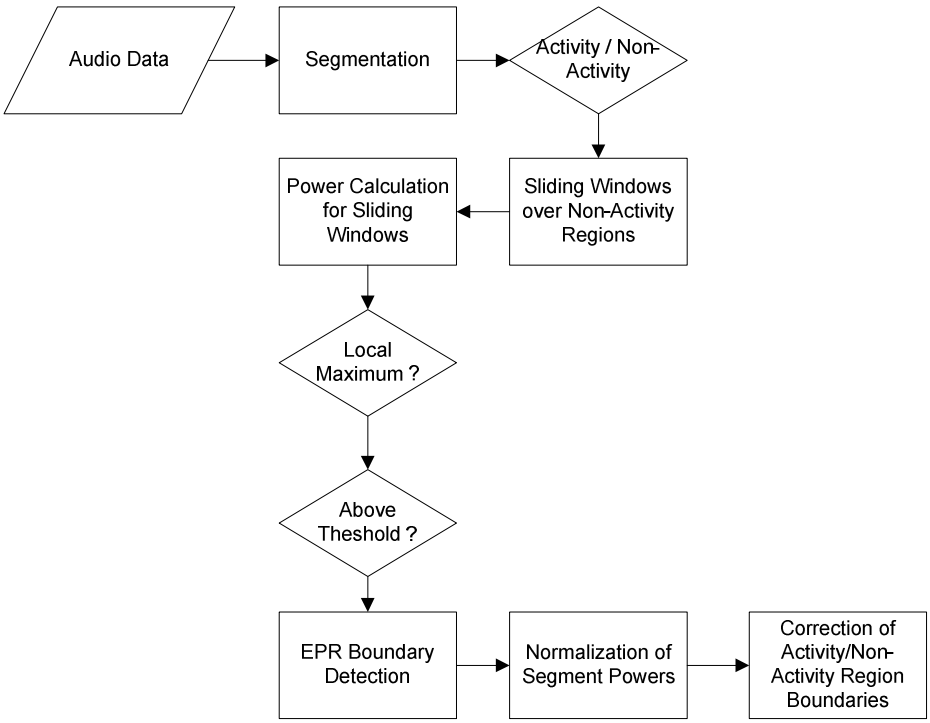


Figure 20: EPR Detection Algorithm

Tests on EPRs show that normalization of segment energies according to EPRs improved both recall and precision of activity regions. Also the cost function described in Equation 8 shows that, EPR normalization improved the test results.

There are two different parameters tested on EPR detection method similar to the UES method. The first one is the length of the sliding windows and the second parameter is the threshold value filtering the local maximum values. The change of precision, recall and cost function with respect to window size and threshold value is given in Figure 21-18. According to the tests, the window length is determined as 5, and the threshold value is 1.5. The comparison of best recall precision and cost function values between applying and not applying the EPR detection method is given in Figure 27.

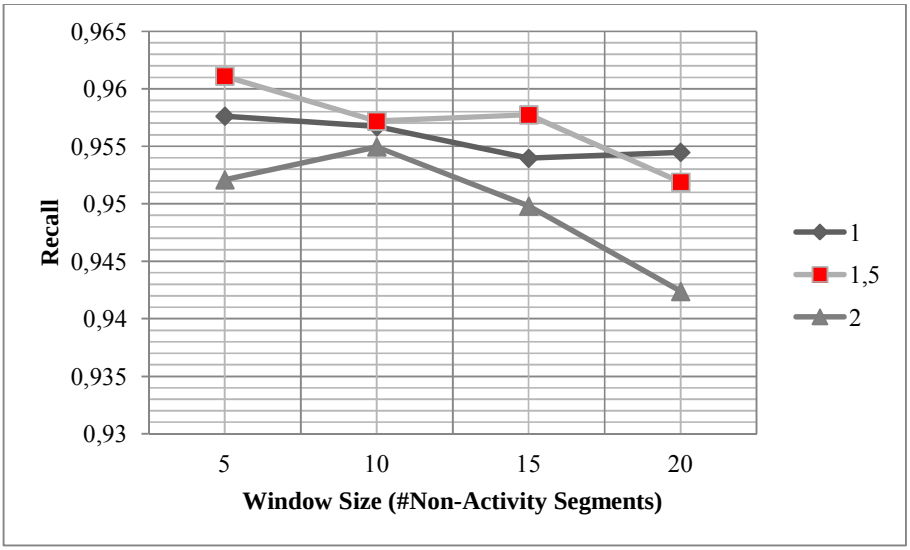


Figure 21: Recall vs Window Size (vs Threshold)

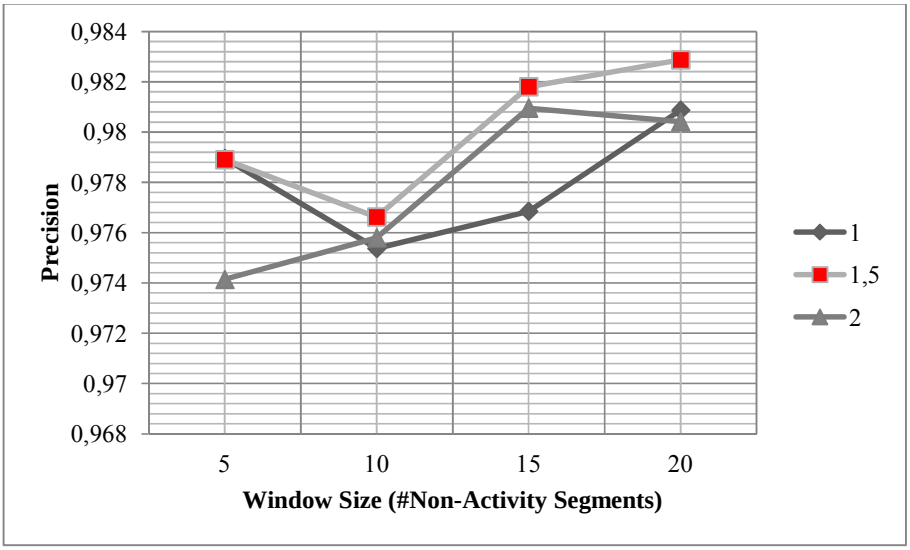


Figure 22: Precision vs Window Size (vs Threshold)

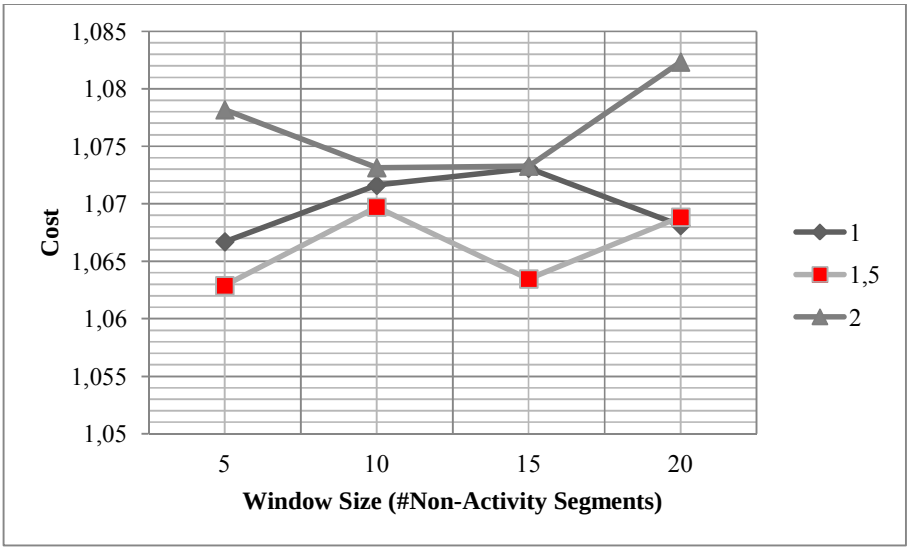


Figure 23: Cost vs Window Size (vs Threshold)

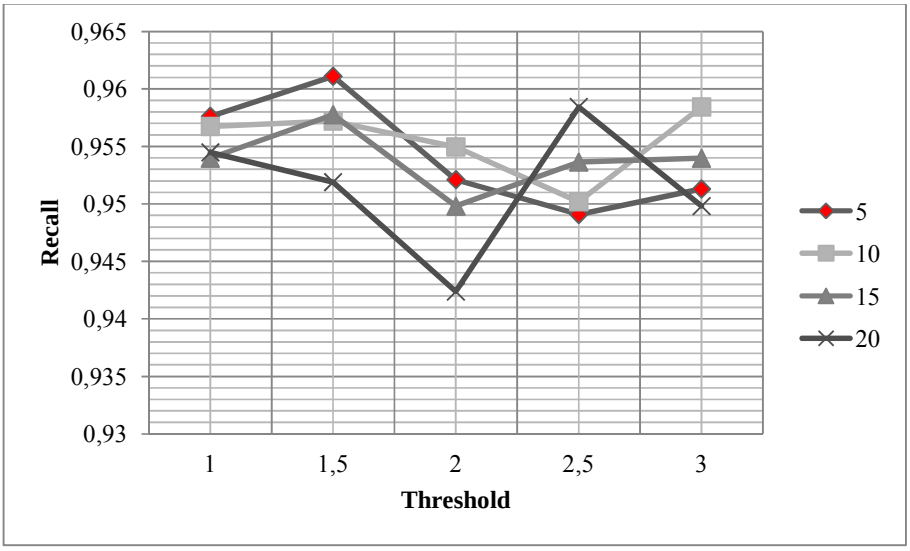


Figure 24 Recall vs Threshold (vs Window Size)

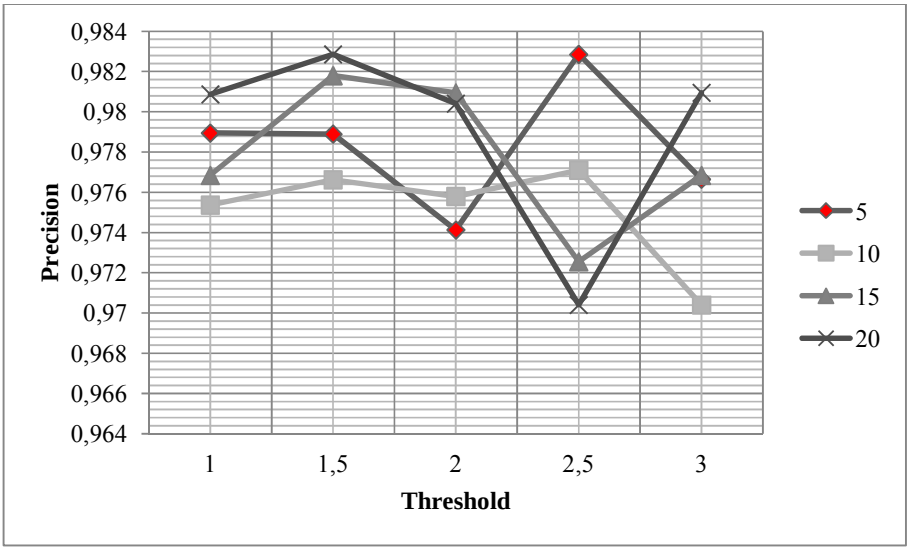


Figure 25: Precision vs Threshold (vs Window)

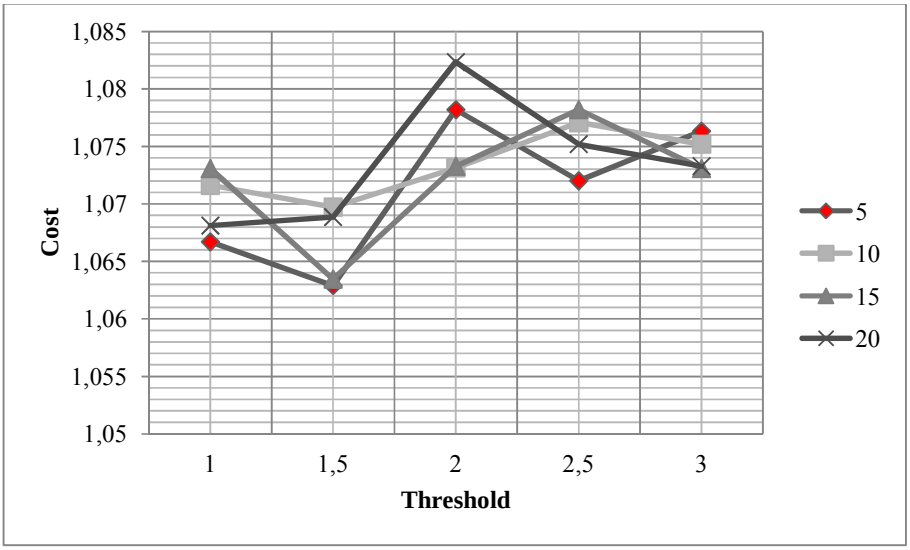


Figure 26: Cost vs Threshold (vs Window)

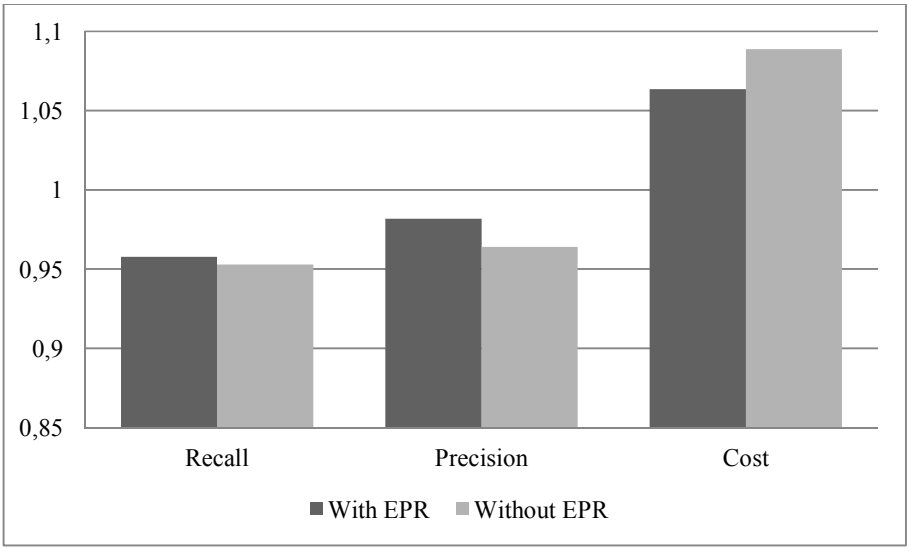


Figure 27: Comparison of Region Detection with EPR and without EPR

CHAPTER 3

FEATURES, FEATURE SELECTION METHODS AND CLASSIFIERS

There are numerous different features in the literature, which are used for different audio pattern recognition operations. In this study, commonly used audio features in the area of audio event detection are selected. All features are extracted using a frame length of 25ms and a frame slide of 10ms. The features and their abbreviations used in the rest of this study are given in Table 1.

Table 1: List of Features and Abbreviations

Mel Frequency Cepstral Coefficients	(MFCC)
Perceptual Linear Prediction	(PLP)
Spectrum Band Power	(SBP)
Spectral Flow Direction	(SFD)
Band Harmonicty	(HRM)
Spectral Roll-off	(SRO)
Zero Crossing Rate	(ZCR)
Spectrum Flatness	(SRF)
Spectrum Centroid	(SRC)

3.1. Audio Features

In this section, the audio features given in Table 1 are described.

3.1.1. Mel Frequency Cepstral Coefficients (MFCC)

Mel-Frequency Cepstral Coefficients are widely used in audio pattern recognition area. Several examples in the literature for the usage of MFCCs in audio event detection problem can be found [9,18,20,33,35,37-40]. The flow diagram of MFCC extraction process is given in Figure 28.

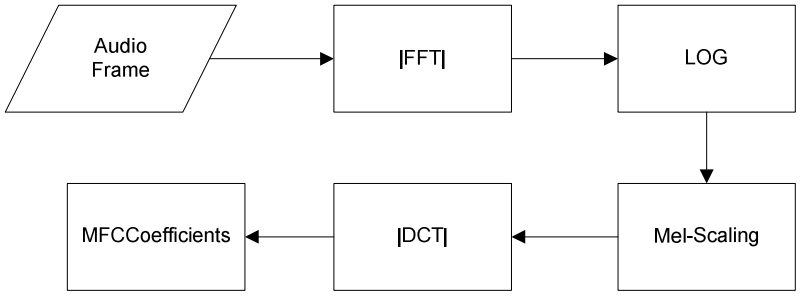


Figure 28: MFCC Flow Diagram

The audio signal is divided into short-time frames, to satisfy stationarity property. Then each audio frame is transformed to frequency domain using Fast Fourier Transform (FFT). The amplitudes of the complex outputs of FFT are calculated and their logarithm is taken. The resulting signal is scaled into Mel-Frequency band. Since human hearing and perception of sounds is non-linear, mel-frequency scale is used to model the human perception on audio signals. Mel-frequency scale is expressed as

$$Mel(f) = \begin{Bmatrix} f & f < 1000\ Hz \\ 2595 \times \log(1 + \frac{f}{700}) & f \geq 1000\ Hz \end{Bmatrix} \quad 9$$

Figure 29 shows the mapping from linear frequency band to the mel-frequency band.

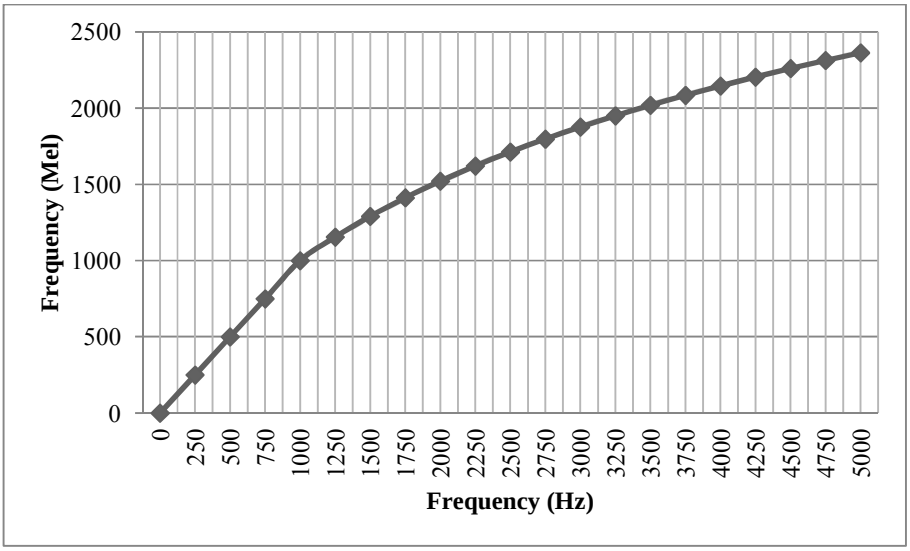


Figure 29: Frequency to Mel-Frequency Mapping

Finally, log-Mel-spectrum signal is converted using Discrete Cosine Transform (DCT). The magnitudes of 13 coefficients of DCT are used as MFCC. The number "13" here is variable, and 13 is the commonly used value in the literature.

3.1.2. Perceptual Linear Prediction (PLP)

PLP parameters aim to model the perceptually relevant parts of the audio signal [23,31-33,41,42]. For this reason, the audio signal is first filtered and pre-emphasized using "perceptual filters". The flow diagram of the extraction of PLP values is given in Figure 30.

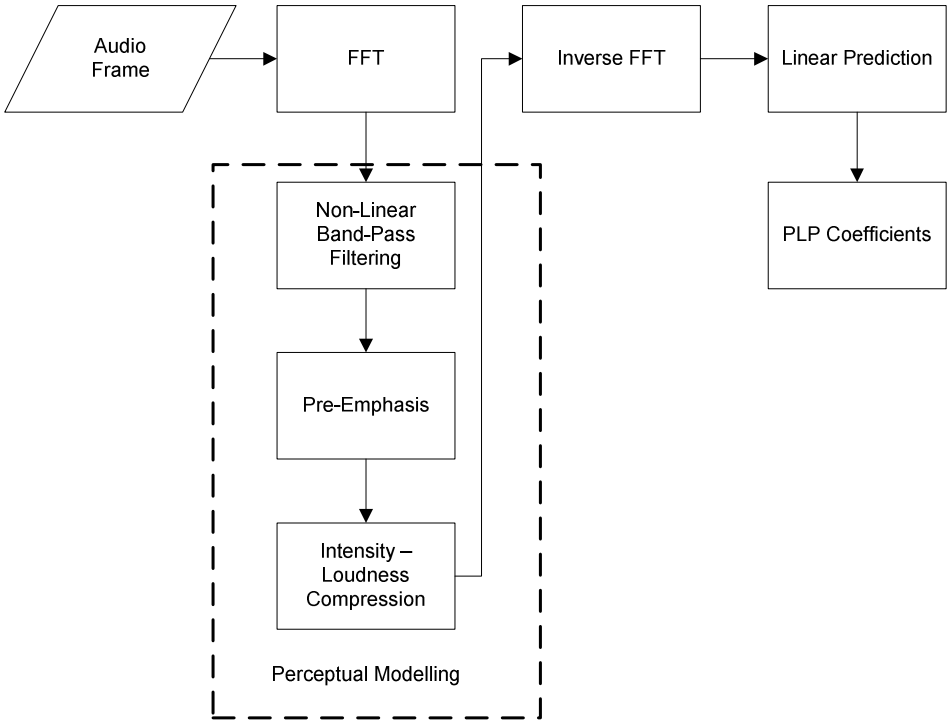


Figure 30: PLP Flow Diagram

The signal is transformed into frequency domain and pre-processing operations are applied. The first operation is the "critical band filtering", where the signal is passed through a non-linear band-pass filter, to model the non-linear frequency warping property of human hearing. Then a pre-emphasizing process is applied and followed by intensity to loudness compression, which models the non-linear human loudness perception. After that the signal is transformed back to the time domain and linear prediction analysis is performed. The coefficients of the all-pole linear prediction model are the PLP values.

3.1.3. Spectrum Band Power (SBP)

SBP describes the powers of signal at different bands of the spectrum. Figure 31 shows the flow diagram of SBP extraction procedure.

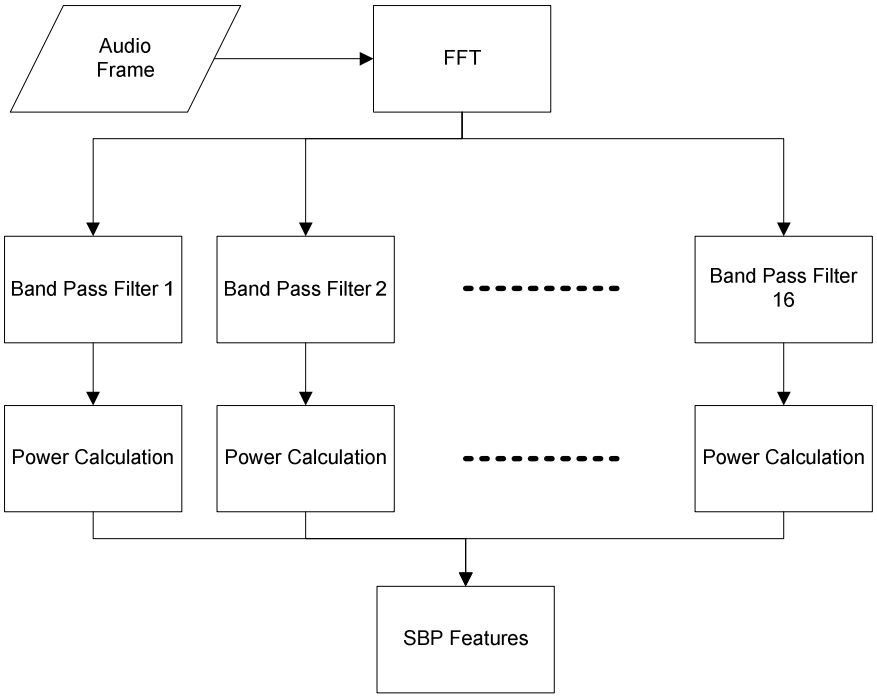


Figure 31: SBP Flow Diagram

The audio frame is transformed into frequency domain by using FFT. Then the signal is passed through several different band-pass filters and the powers of the resulting signals are computed. In this study, the spectrum is divided into 16 bands between the frequencies of 50 Hz and 4000 Hz.

3.1.4. Spectral Flow Direction (SFD)

SFD [9] describes the temporal behaviour of spectrogram in terms of the direction of energy flow in time at the peaks of the spectrum. The maximum point of the cross-correlation between the spectra of two consecutive frames is the SFD value corresponding to the first frame. SFD is calculated as

$$SFD(n) = argmax_l \left(\sum_{i=w}^{\frac{N}{2}-w} s(n,i) \times s(n+1, i+l) \right) \quad l = -w, -w+1 \ldots, w \qquad 10$$

N: FFT Length

l: Amount of lag

s(n,i): Energy at frequency frame n of bin i.

3.1.5. Band Harmonicity (HRM)

HRM [9] describes the harmonic content of the given audio frame. Unlike the common harmonicity measures [18,34,35,43,44], Band Harmonicty does not require fundamental frequency detection. HRM assumes that, for a perfectly harmonic signal, the magnitude of the frequency transform is also periodic with the fundamental frequency and therefore; FFT of the FFT magnitude of a harmonic signal has a maximum point at the fundamental frequency value. Band Harmonicity can be computed over multiple bands, which allows computing different harmonicity values for different bands. Considering the difference of the bandwidths of different audio signals, band-wise harmonicity is a discriminating property. The flow diagram of HRM computation is given in Figure 32.

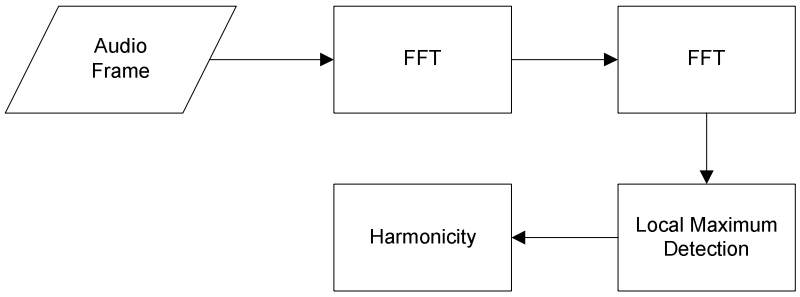


Figure 32: Band Harmonicity Flow Diagram

3.1.6. Spectral Roll-Off

The SRO is the frequency up to which 90% of the signal energy is contained. SRO is calculated as

$$\sum_{i=0}^{f_{ro}} x(i) = \frac{90}{100} \sum_{i=0}^{N} x(i) \qquad 11$$

where x(i) represents the samples in a frame.

3.1.7. Zero-Crossing Rate

ZCR is the number of zero-crossings in time domain, for an audio frame. ZCR is normalized by dividing the number of zero-crossings by the number of samples in an audio frame. ZCR is calculated as

$$ZCR = \frac{1}{N}\sum_{i=1}^{N}\frac{|sgn(i) - sgn(i-1)|}{2} \quad (12)$$

3.1.8. Spectral Flatness (SRF)

Spectral Flatness is a measure of tonal strength of the sound [34,45,46]. A sound with a spectrum of more distinct peaks is said to be more tonal than a sound with a white-noise like, flat spectrum. Like SBP, SRF is also extracted band-wise. The flowchart of SRF extraction method is given in Figure 33. The calculation of spectral flatness is done using the formula given in Equation 13.

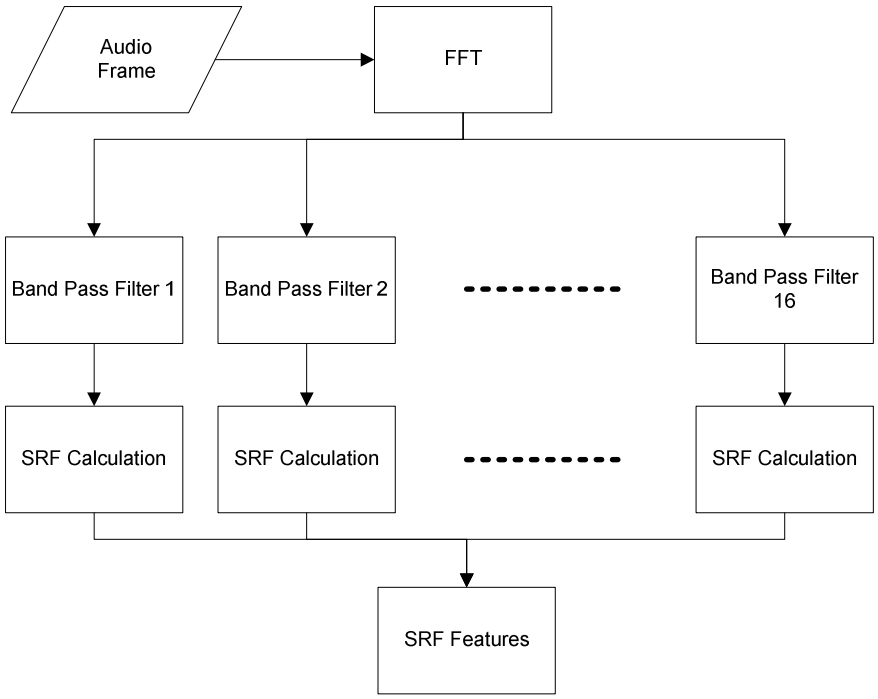


Figure 33: SRF Flow Diagram

$$SRF = \frac{\sqrt[N]{\prod_{i=0}^{N} x(i)}}{\frac{1}{N}\sum_{i=0}^{N} x(i)} \quad (13)$$

3.1.9. Spectrum Centroid (SRC)

Spectrum Centroid is the center of mass of the spectrum. It describes the brightness of the sound. Sounds with higher SRC values are said to be brighter than sounds with lower SRC values. The calculation of SRC is done using the formula given in Equation 14.

$$SRC = \frac{\sum_{i=0}^{N} ix(i)}{\sum_{i=0}^{N} x(i)} \quad 14$$

3.2. Feature Statistics

In this study, audio segments are accepted as the smallest unit of audio events, such that audio event detection on broadcast recordings is based on decisions given per segment. Every segment, having different spectral and temporal characteristics, is described by various audio features. However, these features themselves are not sufficient to describe the whole segment, since each feature is extracted from an audio frame, and segments consist of varying number of frames. For this reason, frame based audio features are used to compute some statistics for each frame, and these statistics are used as new descriptors for the segment itself.

The list of feature statistics used in this study is given in Table 2. These statistics are the most frequently used statistics in the literature for audio event detection and retrieval [17,19,24,43,45,47,49-54].

Table 2: List of Feature Statistics

1. Mean
2. Minimum & Maximum
3. Median
4. Variance
5. Skewness
6. Kurtosis

3.2.1. Mean

Mean of the feature vectors in a segment is the feature vector, having dimensions calculated by taking the average of feature vectors in the segment for that dimension. Mean is

calculated using the formula given in Equation 15. Mean is a vector which coarsely represents all of the vectors in the segment.

$$Mean(i) = \frac{1}{N}\sum_{n=1}^{N} f_n\,(i) \tag{15}$$

N: Number of frames in a segment.
f_n: Feature vector corresponding to the frame n.
i: Feature vector dimension index.

3.2.2. Minimum & Maximum

Minimum and maximum of the feature vectors in a segment is the vector that consists of minimum and maximum feature values of the corresponding dimensions. Minimum and maximum vectors represent the range of the values in a segment, expressing the extreme cases. Minimum and maximum vectors are calculated as

$$\begin{aligned} Minimum(i) &= f_{sort}(f, 0)(i) \\ Maximum(i) &= f_{sort}(f, N)(i) \end{aligned} \tag{16}$$

3.2.3. Median

Median of the feature vectors in a segment is the vector that consists of feature values of the corresponding dimensions which are in the middle when they are ordered. The formula used to calculate median vector is given in Equation 17. Median vector represents the values in the middle, making again a coarse estimation for each vector in the segment, yet making sure of the given value is observed, unlike the mean vector.

$$Median(i) = f_{sort}\left(f, \frac{N}{2}\right)(i) \tag{17}$$

3.2.4. Variance

Variance of the feature vectors in a segment is the vector that consists of the variance values of the corresponding dimensions. The formula used to compute the variance is given in Equation 18. Variance vector represents how far the samples inside a segment are from the mean. The greater the variance, the less similar feature vectors are.

$$Variance(i) = \frac{1}{N-1}\sum_{n=1}^{N}(f_n(i) - Mean(i))^2 \quad (18)$$

3.2.5. Skewness

Skewness of the feature vectors in a segment is the vector that consists of the skewness values of the corresponding dimensions. The formula used to compute the skewness is given in Equation 19. Skewness is the measure of the asymmetry of the probability distribution of given values.

$$Skewness(i) = \frac{\frac{1}{N}\sum_{n=1}^{N}(f_n(i) - Mean(i))^3}{(\frac{1}{N}\sum_{n=1}^{N}(f_n - Mean(i))^2)^{3/2}} \quad (19)$$

3.2.6. Kurtosis

Kurtosis of the feature vectors in a segment is the vector that consists of the kurtosis values of the corresponding dimensions. The formula used to compute the kurtosis is given in Equation 20. Kurtosis measures how "peaked" is the probability distribution of given values.

$$Kurtosis(i) = \frac{\frac{1}{N}\sum_{n=1}^{N}(f_n(i) - Mean(i))^4}{(\frac{1}{N}\sum_{n=1}^{N}(f_n(i) - Mean(i))^2)^2} - 3 \quad (20)$$

3.3. Feature Selection Algorithms

In this chapter, the feature selection algorithms, which are used and compared in this study, are mentioned.

3.3.1. Principal Component Analysis

Principal Component Analysis (PCA) is a basic statistical tool that is used in many applications in pattern recognition area. It is an orthogonal transformation of the possibly correlated feature space onto a new one, where the new features are uncorrelated and with reduced dimensions.

The "principal components" are obtained to deal with the linear dependency among the feature variables [53]. Each principal component is a linear combination of feature variables. The linear combination can be formulated as given in Equation 21.

$$\overrightarrow{PC} = A^T \vec{X} \quad (21)$$

X: feature vector

A: linear factors vector

In PCA, the linear factors for obtaining the principal component are chosen such that the resulting feature variables have maximized variance. Since covariance matrix obtained by the original dataset represents the variance of the variables, eigenvectors of the covariance matrix are computed to form the transformation matrix. The eigenvector with greater eigenvalues tells us that the data set has more variance in that dimension than others. In PCA, the eigenvectors are sorted according to their eigenvalues, and dimensions with small variances (eigenvalues) can be omitted. Since smaller variance in a dimension refers to less information provided by that dimension, the information loss after dimension reduction is minimized.

The computational flow diagram of PCA is given in Figure 34. Firstly the covariance matrix of given variables are computed using the whole dataset without any class labels. Then the eigenvectors and corresponding eigenvalues are computed. Eigenvectors are sorted by the corresponding eigenvalues, and a transformation matrix is formed using the selected eigenvectors. First M eigenvectors are selected, where M is the number of dimensions of the new feature vectors. M is calculated with the formula given in Equation 22.

$$\min_M \left(\frac{\sum_{i=1}^{M} \lambda_i}{\sum_{i=1}^{N} \lambda_i} \right) > 0.95 \quad (22)$$

N: number of original dimensions

M: number of reduced dimensions

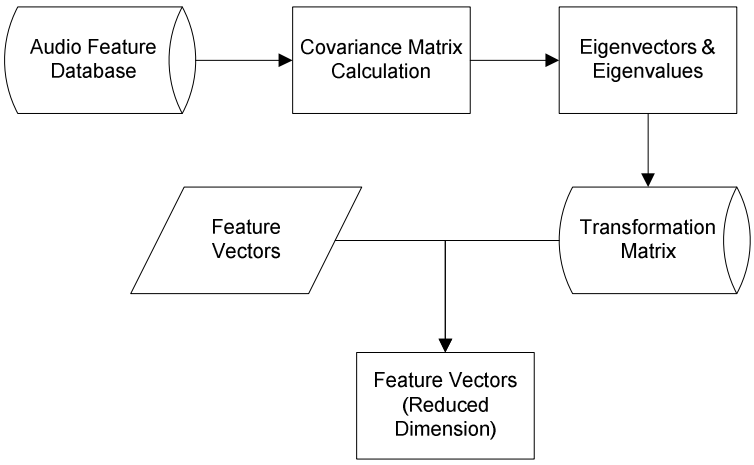


Figure 34: PCA Flow Diagram

3.3.2. Information Gain Ranking

Information Gain (IGR) is another feature selection method that is highly popular in pattern recognition areas [20,54,55]. "Information Gain" (aka Kullback-Leibler divergence) is used to measure the similarity between two distributions. IG can be described as the change of two entropies H(C), the entropy of class C, and H(C|X), the entropy of class C when the sample X is given. The decrease observed in the entropy after the sample X is given is perceived as the information provided by the sample X. The formula used to calculate IG is given as

$$H(C) = -\sum_{c \in C} p(c) \log_2 p(c) \qquad 23$$

$$H(C|X) = -\sum_{x \in X} p(x) \sum_{c \in C} p(c|x) \log_2 p(c|x)$$

$$IG = H(C) - H(C|X)$$

IG can also be defined as a measure showing how well a variable discriminates between two classes. IGR is the method of selecting features sorted by their information gain. Higher IG value corresponds to more information.

3.3.3. Chi-Square Ranking

Chi-Square distribution with *k* degree of freedom can be defined as the distribution of the sum of squares of *k* independent standard normal Gaussian random variables [56-58]. Chi-

Square Ranking method measures the Chi-Square statistics of each variable, and sorts them according to these measures. Chi-Square statistics measure can be calculated using the formula given in Equation 24.

$$\chi^2 = \sum_{i=1}^{m}\sum_{j=1}^{2}\left(\frac{A_{i,j} - \frac{R_i C_j}{N}}{\frac{R_i C_j}{N}}\right) \quad 24$$

m: number of intervals

R_i: number of features in the interval i.

C_j: number of features in the class j.

A_{ij}: number of features in the interval i, class j.

N: total number of patterns.

In statistics, chi-square method is used to test the independence of two events. In the case of feature selection, these two events can be replaced with the occurrence of the sample and the class. Since chi-square measures the deviation of expected counts from observed counts, a high value of chi-square indicates that the events of the occurrence of the sample and occurrence of the class are dependent. So if these two events are dependent, one can say that occurrence of the sample makes the occurrence of the event more likely [58].

3.4. Classifiers

In this chapter, the classifiers, which are used and compared in this study, are mentioned.

3.4.1. Gaussian Mixture Models

Gaussian Mixture Models (GMMs) are widely used in audio pattern recognition area [1,2,9,16,22-27,32,40,55,59-61]. GMM is not a classifier itself, yet it can be defined as a probabilistic model tool, which uses one or more Gaussians to form a probabilistic model using the information given with the feature vectors. GMMs can be used as a classifier by selecting the most suitable class model for a given sample.

Gaussian Density Function

In GMM, data samples are assumed to be distributed with a Gaussian distribution. Samples belonging to a class do not have to form a single Gaussian, but may have a distribution obtained by a weighted sum of multiple Gaussians. Gaussian density function formula is given in Equation 25. In this case Gaussians are assumed to be multivariate.

$$f(x) = \left(\frac{1}{2\pi}\right)^{\frac{p}{2}} |\sum|^{-\frac{1}{2}} exp\left\{-\frac{1}{2}[(x-m)^T \sum^{-1}(x-m)]\right\} \quad 25$$

For multivariate GMMs, the unknown parameters are the mixture weights, mean vectors and covariance matrices corresponding to each mixture. These parameters have to be estimated in order to create a probabilistic model using the given data samples. This estimation is done by an algorithm called expectation maximization (EM) [61]. EM algorithm is used to find the parameters of GMM, which are weights, mean vectors and covariance matrices for each mixture model that satisfies the maximum likelihood assumption. EM is an iterative algorithm that consists of two steps, expectation and maximization. In the expectation step, initial model parameters are tested with given data samples and a posterior probability is computed. And with the posterior probability values obtained in the expectation step, new parameter values are calculated in the maximization step. This iteration continues until the parameters converge. In GMM classification, for each model obtained, a likelihood value is computed by each sample. The class with greater likelihood value is selected.

3.4.2. Support Vector Machines

Support Vector Machine (SVM) is the second type of classifier that is used in this study. SVMs also have a wide range of application area in pattern recognition [2,19,23,25,35,48,54,62-68]. SVMs can be used for binary or multi-class classification and regression applications. In this study SVMs are used as binary classifiers.

SVM classifier aims to define a linear discriminant function that separates two classes with a maximized margin. This margin can be defined as the minimum distance between the separating hyper-plane and the nearest sample. The relation between hyper-plane and feature vectors can be improved by mapping the feature vectors to a new vector space non-linearly. This mapping requires the calculation of dot-products of the new feature vectors, which can be computationally expensive. Instead, “kernel” functions are used to calculate these dot-products, and obtain non-linear hyper-plane functions. In this study, radial basis kernel function (RBF) given in Equation 26 is used.

$$k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \quad 26$$

CHAPTER 4

EXPERIMENTS

The proposed method in this study is supported with experiments, providing the results for the given algorithm. In these experiments, a large set of audio data is collected, using the TV broadcast recordings. The most frequent events in these recordings are selected as the events to be detected. The features and dimension reduction algorithms which are mentioned above are tested and best feature sets are selected for each audio event class. The proposed algorithm is run with different classifiers which are also mentioned in previous chapter, and the results of these tests are given comparatively. In this chapter, first; information about the training and test dataset is given. Then the methodology of the experiments is explained and finally the results of each experiment are presented.

4.1. The Data Set

The Data Set used in the experiments part of this study consists of samples collected from TV broadcasts within the scope of the KAVTAN, a project for the semantic classification of mass broadcast media, which is developed for RTUK (Radio and Television Supreme Council). The whole data is manually annotated by using the software Praat [69]. A total of 4 hour-long data is annotated by hand and used in training and a total of 18 hours of data is used in test steps of this study. The details of the data set are given in Table 3.

Table 3: Test Set Properties

	Training		Test	
	Duration (Sec)	*# samples*	*Duration (Sec)*	*# samples*
Applause	936,26	2004	50,06	60
Bird	339,02	960	534,55	581
Brake	392,17	752	96,99	71
Cat	281,58	670	45,66	76
Crowd	231,28	270	2203,76	1531
Cry	834,51	2410	183,75	234
Dog	352,65	1128	67,48	106
Explosion	912,66	1458	671,76	833
Gun	985,74	2303	122,30	165
Laughter	430,76	1137	147,70	211
Music	863,85	1913	28338,30	37096
Other	NA	NA	14198,40	11090
Scream	857,55	1910	424,18	508
Sex	704,69	1955	54,31	86
Sing	235,64	640	2530,59	1743
Siren	1249,48	2431	213,22	173
Speech	1072,97	748	13790,90	19785
Water	1112,92	1272	705,93	1018
Total	**11793,73**	**23961**	**64379,84**	**75367**

17 events given in Table 3 are the most frequent audio events which are observed in the data set. The “Other” event in the test set represents unclassified events including silence and noise. Some of these events like speech and music are highly popular in audio event detection studies, whereas cat, dog or sexual sounds are rare to observe. In these experiments, it is aimed to show that, the proposed method for event detection on TV broadcasts is successful for a wide selection of events.

Waveform and spectrogram examples for these events are given from Figure 35 to Figure 40.

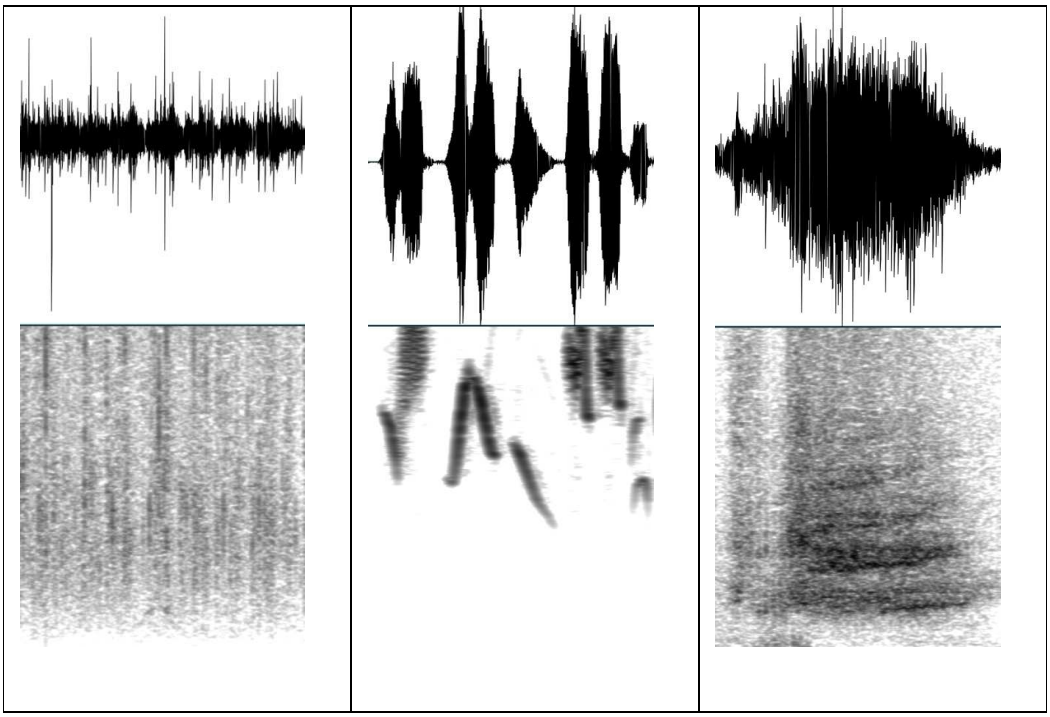


Figure 35: Applause, Bird and Brake Sounds

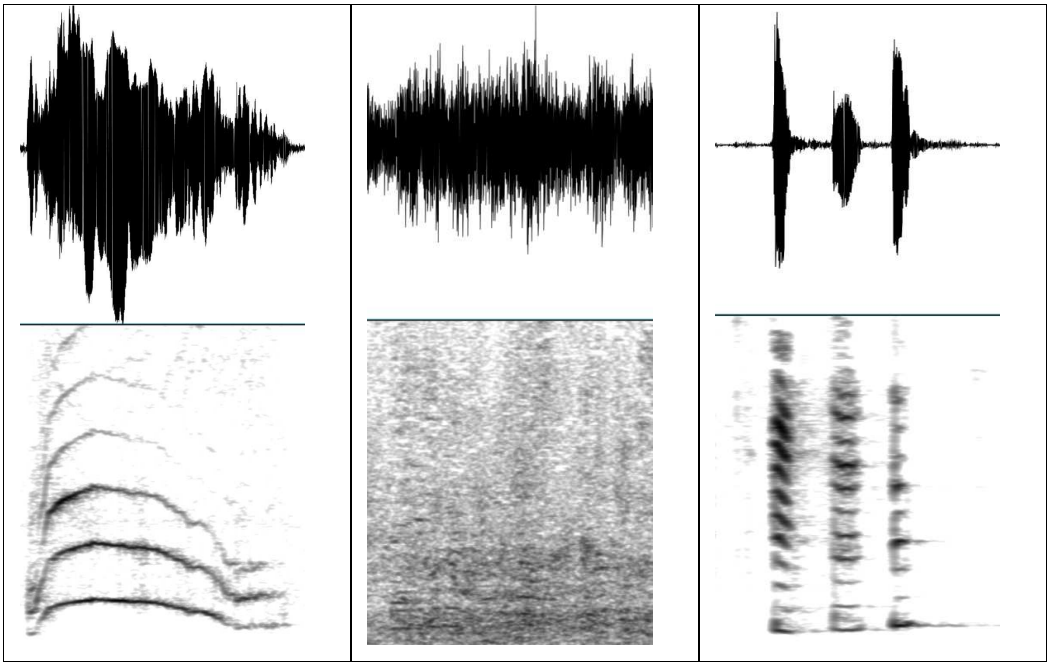


Figure 36: Cat, Crowd and Cry Sounds

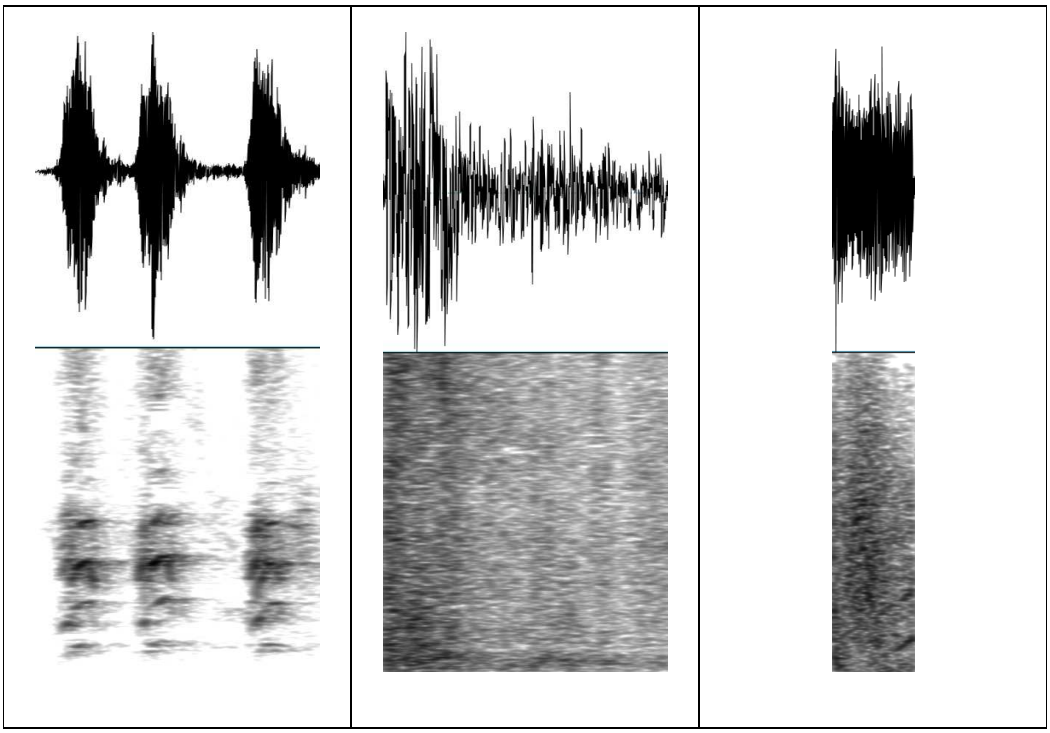


Figure 37: Dog, Explosion and Gun Sounds

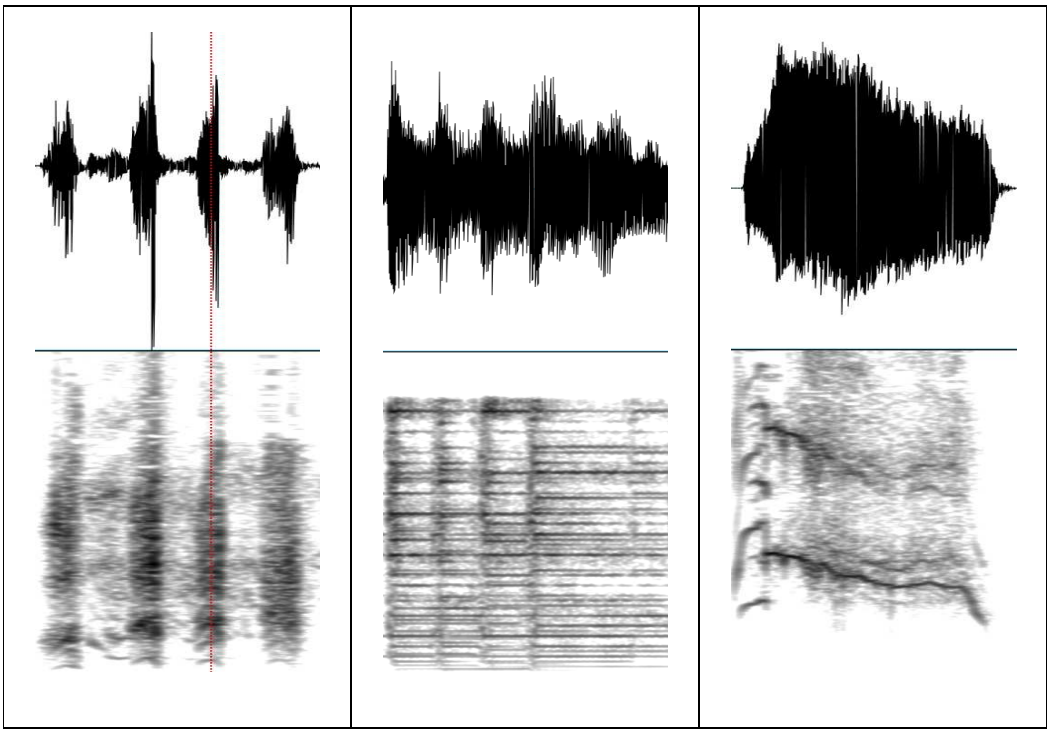


Figure 38: Laughter, Music and Scream Sounds

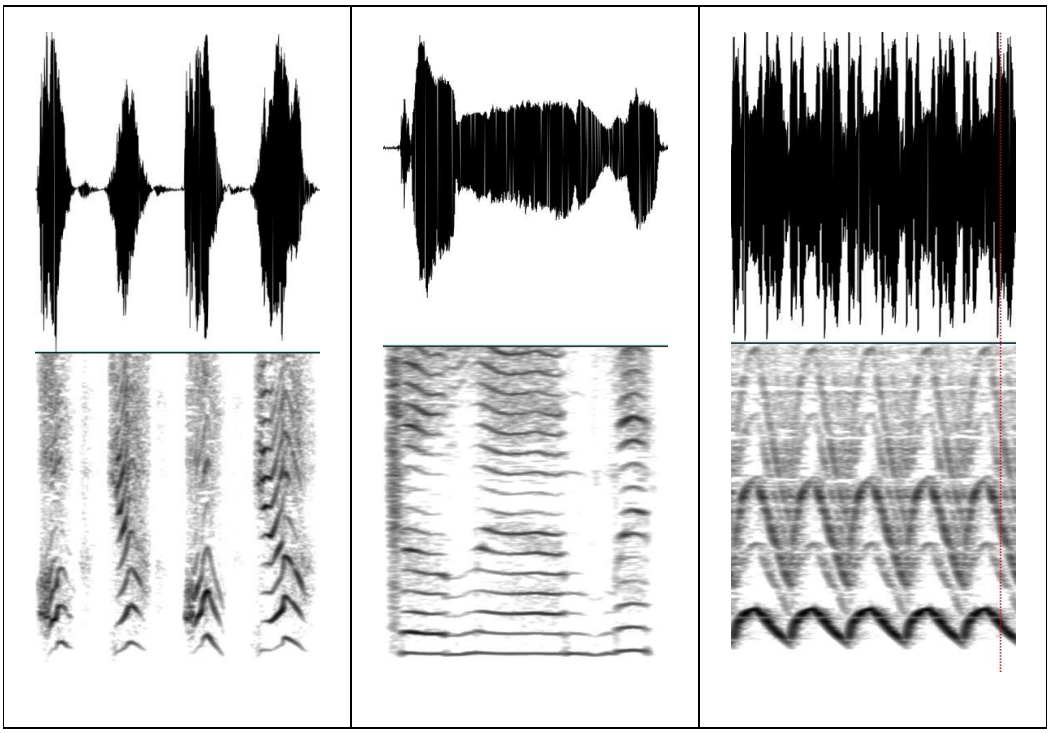


Figure 39: Sex, Singing and Siren Sounds

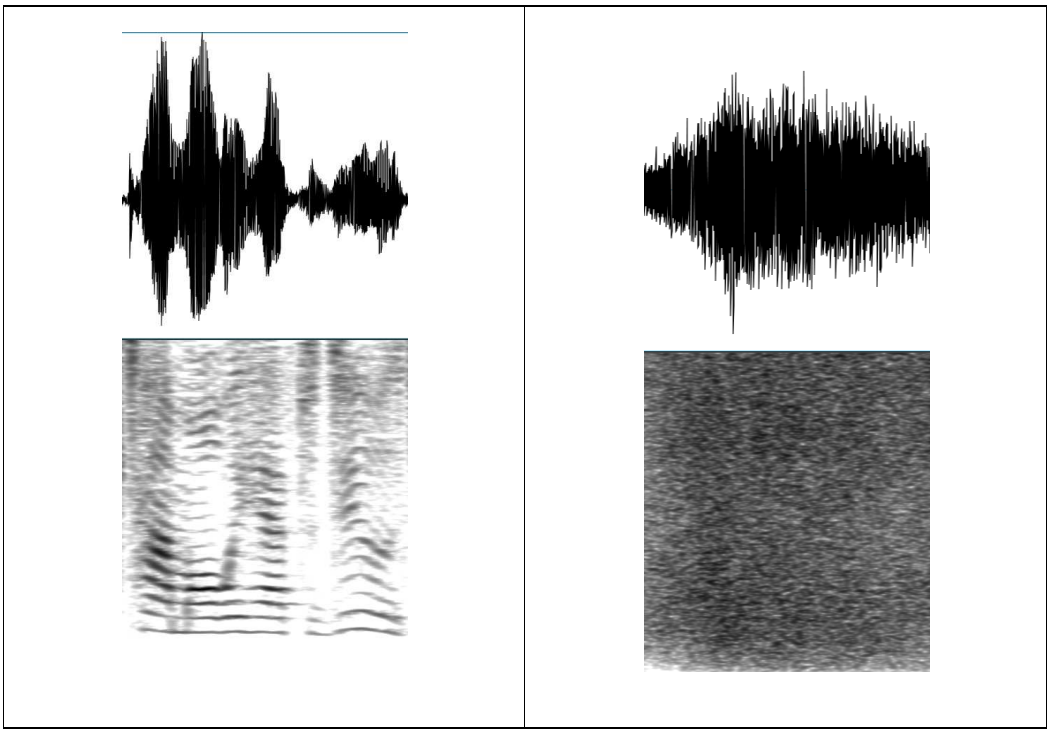


Figure 40: Speech and Water Sounds

In the spectrograms above, given audio events are shown to have a different characteristic and these characteristics are aimed to be represented using various audio features mentioned in section 3. Just looking at the waveforms and spectrograms one can divide those events into two sub classes. Considering harmonicity; music, speech, singing and siren sounds can be thought of audio events with high harmonicity, while explosion, gun-shot and water sounds are harmonic. Gun-shots, cry and sex sounds are shown to be impulsive sounds, where music and scream sounds do not change in time very fast. Scream and bird sounds are shown to have strong components of high frequencies, while explosion and dog sounds have a lower centre frequency value.

4.2. Experiment Methodology

In this study, the best feature set and the best classifier, which discriminates each audio event class, presenting the best performance measure, is aimed to be discovered. The performance of the classification is measured using the two measures which are most frequently used in the literature; the recall and the precision. The definitions and formulations of these measures are given below.

$$Recall = \frac{R_{Relevant} \cap R_{Retrieved}}{R_{Relevant}} \quad (27)$$

$$Precision = \frac{R_{Relevant} \cap R_{Retrieved}}{R_{Retrieved}} \quad (28)$$

Two sets of data are formed for each event, first consisting of the event samples and the second consisting of the samples of other events, which is called the negative set. For each audio event class, a negative event class is formed, randomly mixing the samples of the remaining audio event classes. Considering the unbalanced distribution of the audio events in broadcast, this method is chosen to satisfy the balance in the test sets. Since the precision and recall measures both depend on the number of samples that is being evaluated, the number of samples which belong to an audio event must be in the same order, preferably equal to the number of samples that does not belong to that audio event.

For example, considering an event A with a frequency of 1sn per minute, meaning this event is encountered for 1 second every minute on the average, an audio stream consists of

samples belonging to class A for 1 minute and the stream has samples not belonging to class A for 59 minutes. Assuming that the classifier in hand has an error rate of 1 sample per 58 samples, which can be considered very successful, this classifier is going to yield to a precision rate of 50% which can be considered as almost random. To avoid these kinds of deceptive results, two data sets; one for positive and one for negative samples, are formed and the tests are performed on these datasets. These datasets have exactly the same number of samples, satisfying the condition that the tests are balanced and recall and precision measures are meaningful to determine the success of a classification.

4.3. Experimental Results

The experimental results presented below are obtained using the data set given in Table 3, and best results are selected according to the performance metric F1, given in 29.

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{29}$$

The best recall, precision and F1 scores for each class are given in Table 4. Figure 41 shows the change of F1 among classes, Figure 42 shows the change of recall and Figure 43 shows the change of precision.

Table 4: Best Results

Event	Feature	Classifier	Recall	Precision	F1
Applause	ig	svm	0,890	0,783	0,833
Bird	ig	svm	0,921	0,962	0,940
Brake	chi	gmm	0,798	0,853	0,825
Cat	chi	svm	0,838	0,990	0,907
Crowd	ig	gmm	0,850	0,877	0,863
Cry	chi	svm	0,911	0,669	0,772
Dog	ig	svm	0,901	0,950	0,925
Explosion	ig	svm	0,936	0,973	0,954
Gun	chi	svm	0,852	0,688	0,761
Laughter	chi	gmm	0,822	0,752	0,785
Music	chi	svm	0,933	0,810	0,867
Scream	chi	svm	0,910	0,715	0,800
Sex	pca	svm	0,876	0,763	0,816
Sing	pca	gmm	0,742	0,925	0,823
Siren	ig	gmm	0,957	0,685	0,798
Speech	ig	svm	0,971	0,991	0,980
Water	ig	svm	0,906	0,976	0,939

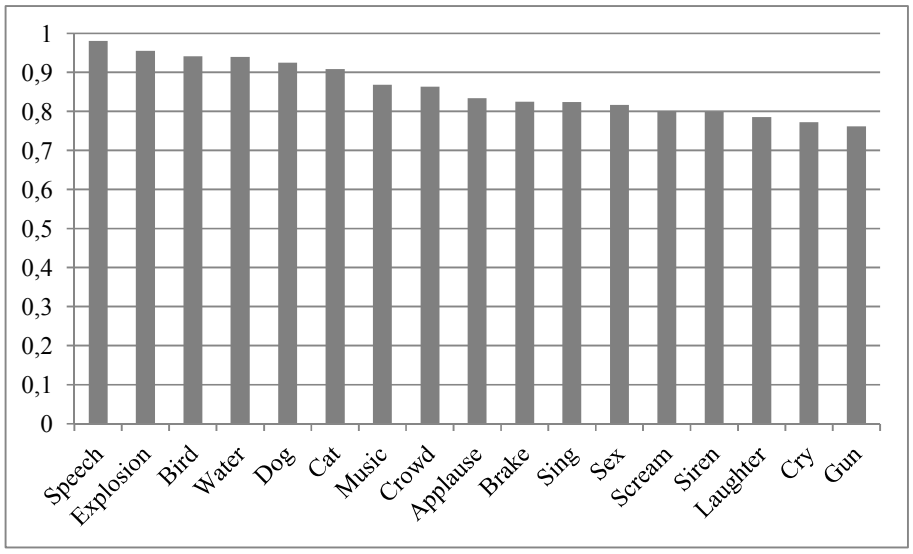


Figure 41: F1 Scores for Each Class

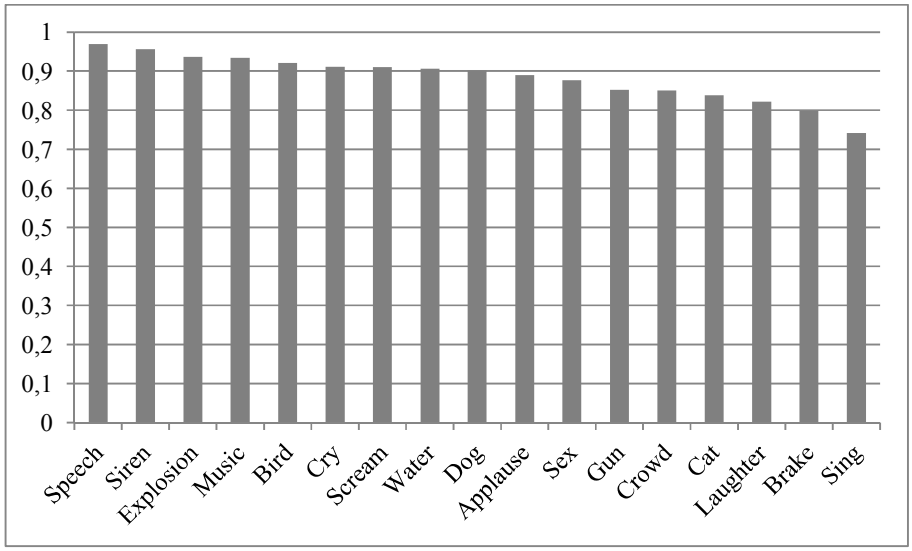


Figure 42: Recall Values for Each Class

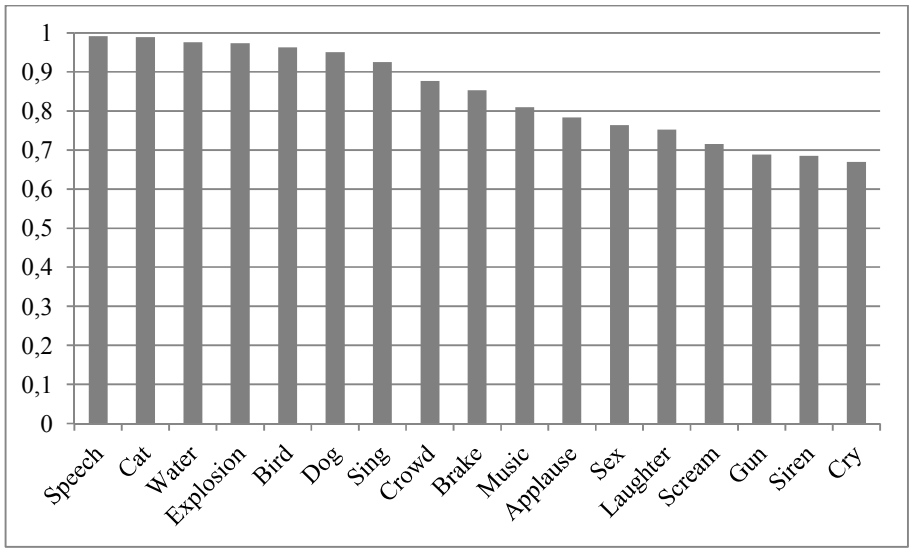
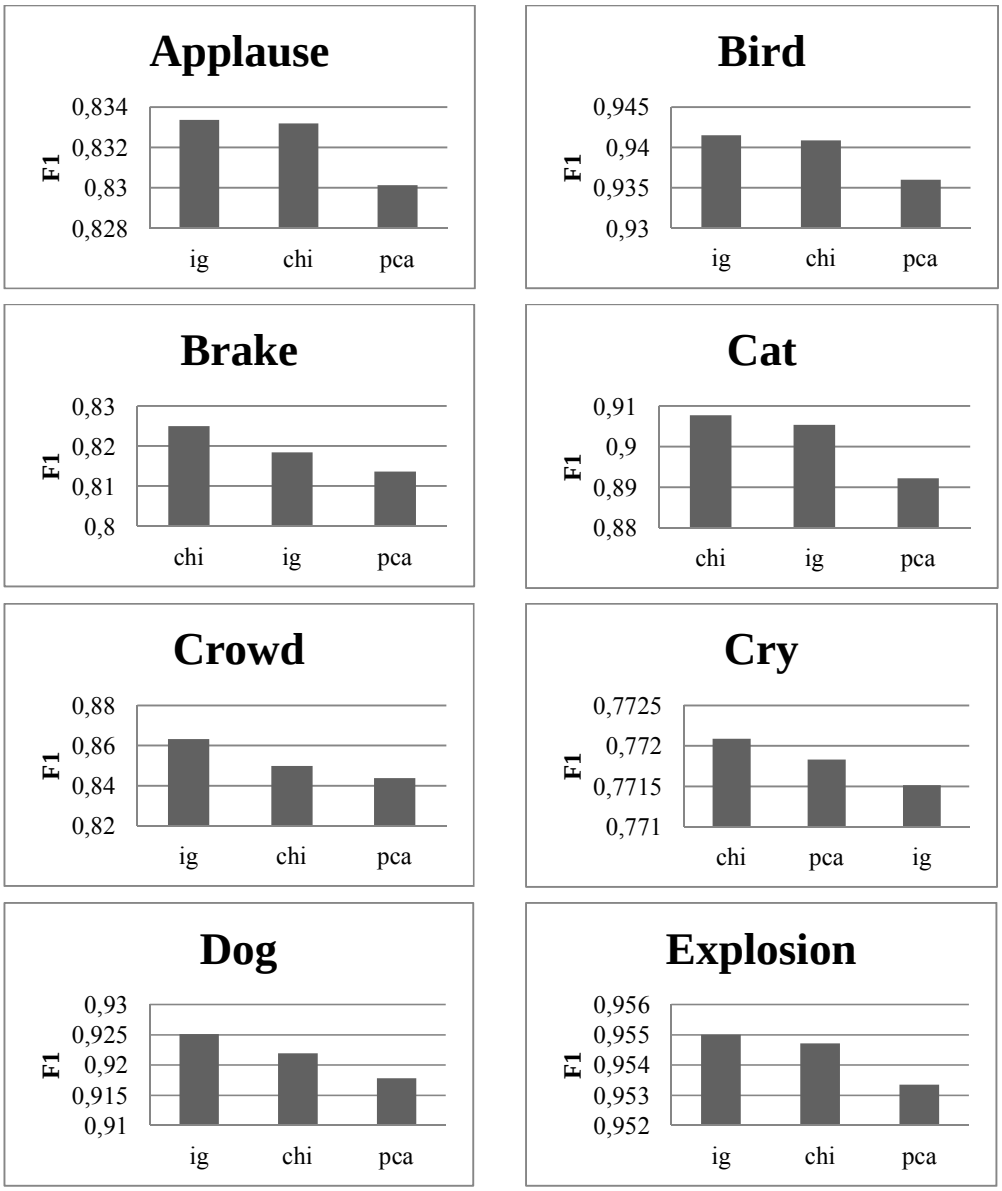


Figure 43: Precision Values for Each Class

As it is shown on these figures, the proposed method has a stable success rate for all different audio events, changing in a range of a maximum F1 of 98% and minimum F1 of 76%. The change of F1 scores, between the most and the least successful events, is found to be 22%. The event with the minimum recall has a value of 74% and the event with the

minimum precision has a value of 67%. Comparing these results with the results given in the literature, the proposed method is fairly successful.

The feature selection and dimension reduction methods are compared and the results are presented in Figure 44.



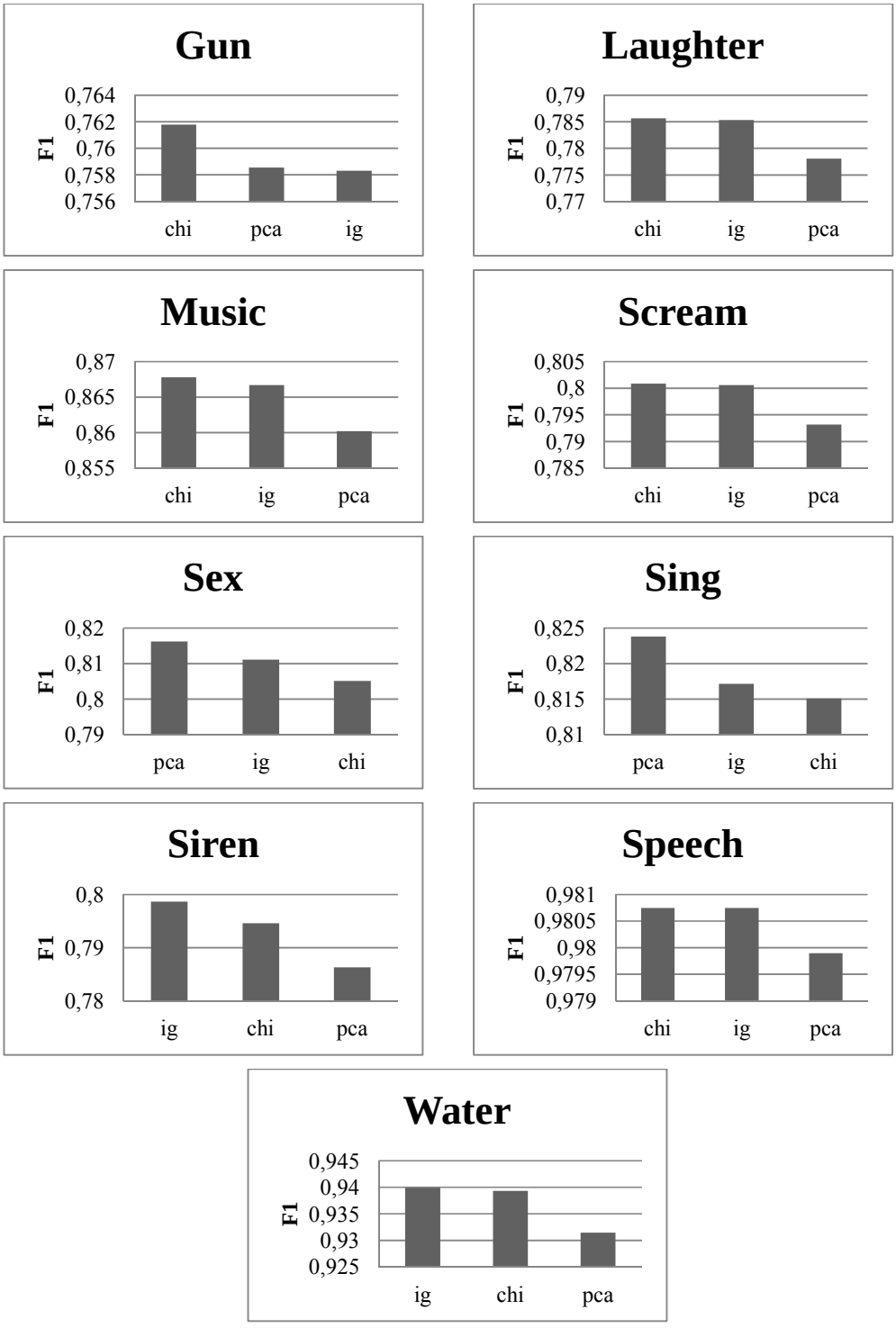
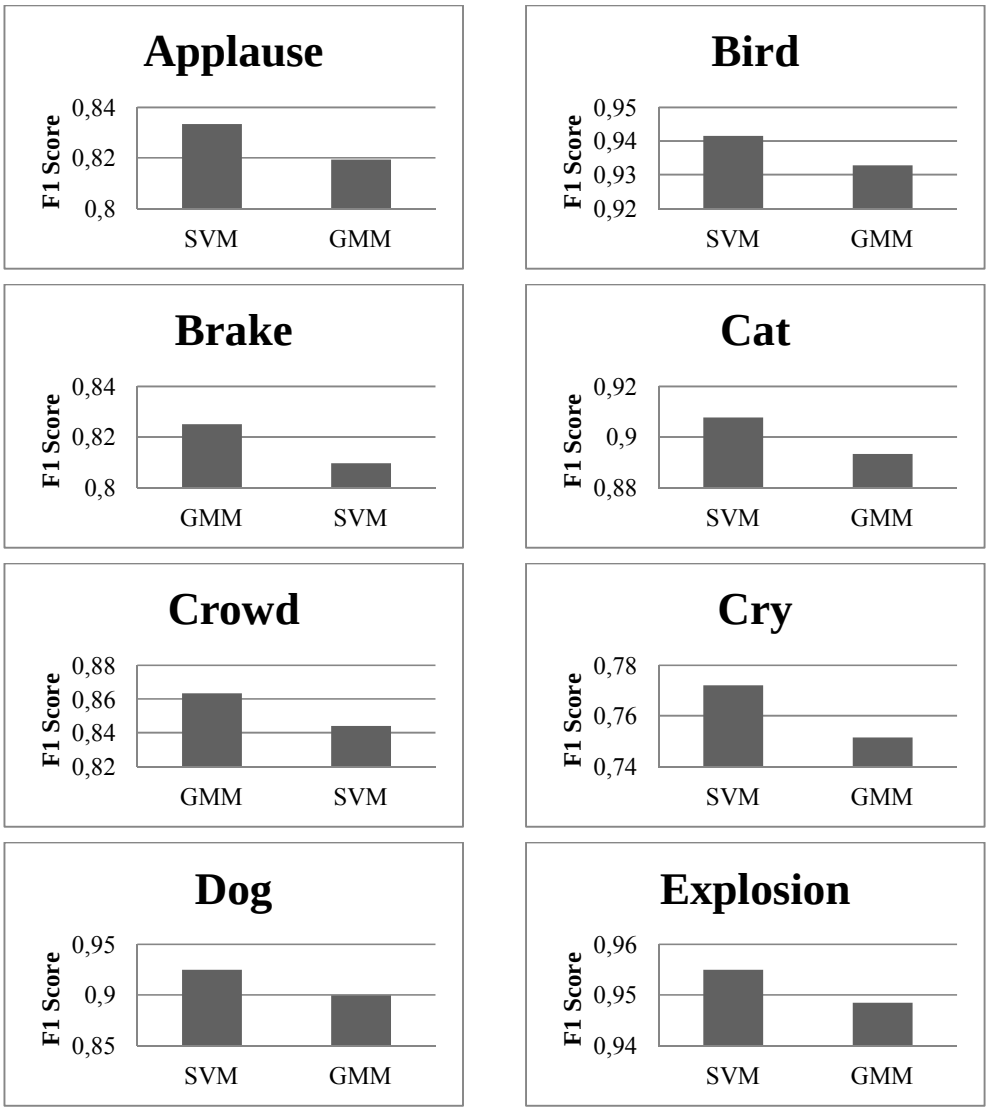


Figure 44: Feature Selection Method Performances for Each Event

As it is shown on the figure above, the performance of the feature selection methods differ very slightly, which indicates the features are eliminated according to the best possible discriminating way. Either feature selection method supplies the very best set of features, which is sufficient to classify these audio events.

Two tested classifiers are also compared and result for each event is given in Figure 45.



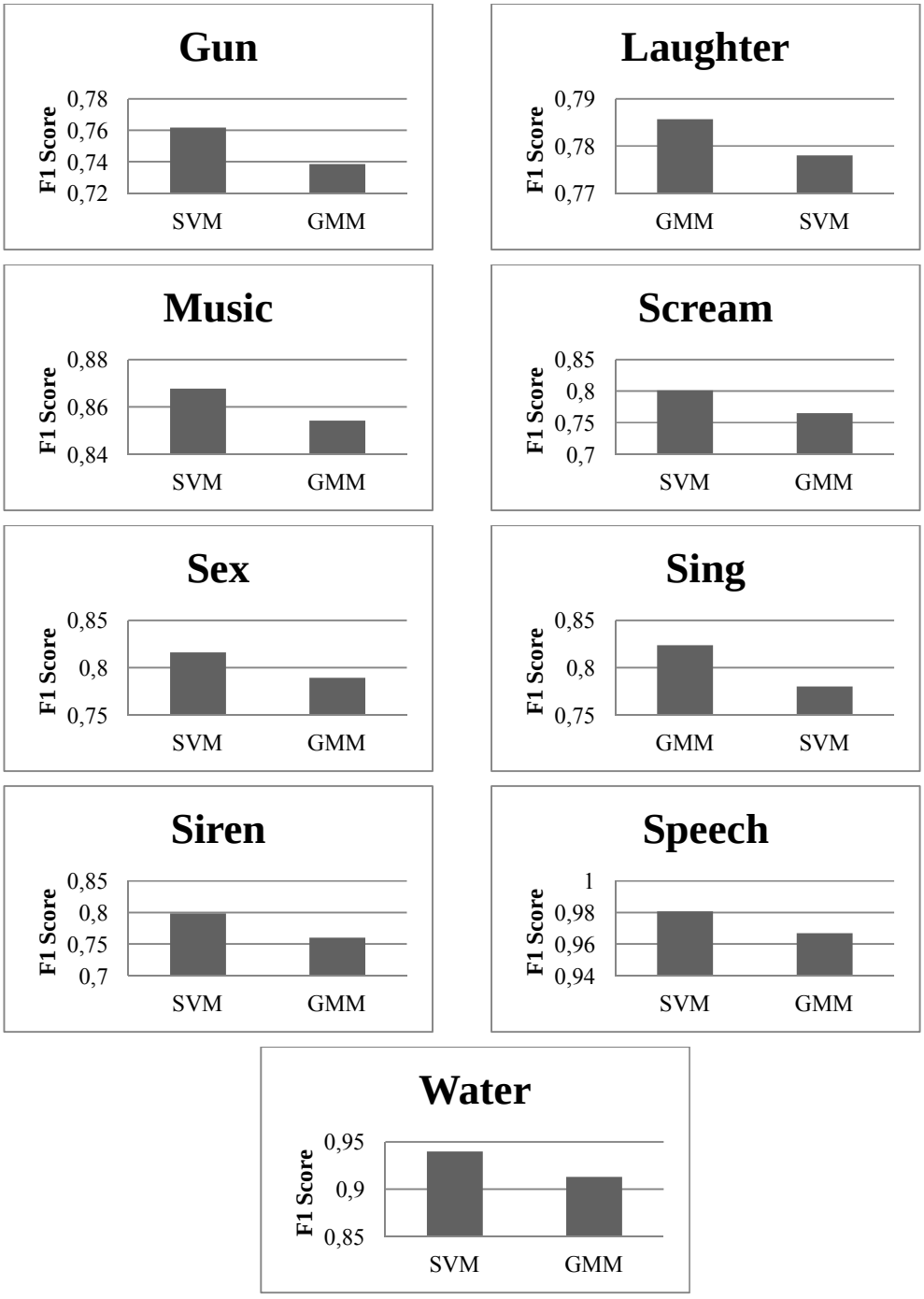


Figure 45: Classifier Performances for Each Event

4.4. Discussion

Comparative results of the audio segmentation algorithm are given in section 2.3, and the advantage of the proposed method is given in Figure 14. In this graph it is shown that the

proposed UES method has a cost twice lower than current BIC method, and the combination of these two methods even increases the performance. The complexity of proposed algorithm is shown to be lower than current BIC method, which decreases the computation time. The computation times of two methods and their combination are given in Figure 15.

Comparing the results of audio event detection algorithm with the results in literature, the proposed method is quite promising. Pfeiffer et al. [15] achieved recall rates of 81% for gunshot, 51% for cry and 93% for explosion sounds. Comparing with the results of the proposed method, the proposed method achieves a recall rate of 94% for explosion, 85% for gun-shots, and 91% for cry sounds. Considering that Pfeiffer used isolated data, this result is promising.

Cai et al. [22] achieved a recall value around 90% for crowd, applause and laughter sounds. The proposed method achieved a recall of 87% on the average, which is a comparable. Portleo et al. [23] tested bird, vehicle, sirens and water sounds, achieving average recall rate of 45%. The result of the proposed method is quite better than those, with a minimum recall of %79.

In [16], an average recall rate of 80% is achieved for speech, music and environmental sounds. In [18], Li et al. tested for music, speech and speech and music together, achieving a result of 90% on the average. Comparing with this result, the proposed method has a recall rate of 89% for speech music and singing on the average. In [19], 16 different audio events, similar to the ones used in this study, are tested with an average recall rate of 89%.

The proposed method uses a collection of features and feature statistics that are widely used in the literature, and using feature selection algorithms, adaptation of the feature set for many different kinds of events is performed automatically. Different feature selection algorithms resulting with similar precision and recall rates for different events also verify that the proposed solution is valid and the selected features describes the selected events successfully.

CHAPTER 5

CONCLUSION

This chapter begins with a summary of the work presented in this thesis. The final section provides a discussion of possible improvements to the methods presented.

5.1. Summary

This thesis presents a method for detecting different audio events which are most frequently observed in audio broadcast. The method involves an audio segmentation stage which divides the audio data into small "homogeneous" segments and enables the detection of event boundaries. These segments are also classified as "Activity" and "Non-Activity" regions, in order to filter out the regions without any aural information. In the event detection stage, remaining segments are classified into several predefined classes by combining many different features and their statistical properties. Using feature selection and dimension reduction methods, this feature set is reduced to a suitable dimension for classifiers. Three different kinds of feature selection and dimension reduction algorithms are tested and compared; and the best results obtained for each audio class is presented. SVMs and GMMs, which are the two most common preferred classifiers in the literature, are used as classifiers in these tests. The proposed system achieved an average recall rate of 88% for 17 different audio events. Compared with the results in the literature, the proposed method has a promising success.

5.2. Future Work

In this study, an "audio segmentation + detection" approach is followed. This idea seems advantageous compared with the methods which segment and classify audio at the same time, such as methods based on Hidden Markov Models, since the segment boundaries are predefined. However, no performance comparisons are given in this study. As a future work, the proposed method can be compared with HMM-based methods.

In the proposed method, audio segments are presented with a single feature vector, which is computed using the statistics of a feature set which is computed over this segment. Another

future work is to use HMMs to classify the whole segment instead of feature statistics. The temporal behaviour of audio signal within a segment, if modelled successfully, can increase the obtained performance.

Observing the given events and their distribution among the whole data, one can say that the existence of some events is correlated with each other. For example, it is much likely to find an explosion segment near a gun-shot segment, or music segments tend to appear consecutively, a single music segment among other kinds of events is rare. If exists and taken into account, these temporal distribution properties and correlation of events with each other can increase the detection performance.

Finally in this study, events are aimed to be modelled using a positive training set consisting of samples belonging to that event and a negative training set consisting of samples which do not belong to that event. As a future work, negative set can be divided into sub-classes and classification can be done by using a combination of sub-classifiers, in order to describe the feature space better.

REFERENCES

[1] L. Lu, H. Jiang, and H. Zhang, "A robust Audio Classification and Segmentation Method," *ACM International Conference on Multimedia*, 2001, p. 203.

[2] L. Lu, H.-J. Zhang, and S.Z. Li, "Content-Based Audio Classification and Segmentation by Using Support Vector Machines," *Multimedia Systems*, vol. 8, Apr. 2003, pp. 482-492.

[3] L. Lu, R. Cai, and A. Hanjalic, "Audio Elements Based Auditory Scene Segmentation," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Ieee, 2006, p. V-17-V-20.

[4] R. Cai, L. Lu, and A. Hanjalic, "Unsupervised Content Discovery in Composite Audio," *ACM International Conference on Multimedia*, 2005, p. 628.

[5] M.M. Goodwin and J. Laroche, "Audio Segmentation By Feature-Space Clustering Using Linear Discriminant Analysis And Dynamic Programming," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, IEEE, 2003, p. 131–134.

[6] S. Pfeiffer, "Pause Concepts for Audio Segmentation at Different Semantic Levels," *ACM international conference on Multimedia*, 2001, p. 187.

[7] J. Foote, "Automatic Audio Segmentation Using A Measure Of Audio Novelty," *IEEE International Conference on Multimedia & Expo (ICME)*, IEEE, 2000, p. 452–455.

[8] G. Tzanetakis and P. Cook, "Multifeature Audio Segmentation for Browsing and Annotation," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, IEEE, 1999, p. 103–106.

[9] U. Zubari, E.C. Ozan, B.O. Acar, T. Ciloglu, E. Esen, T.K. Ateş, and D.O. Onur, "Speech Detection On Broadcast Audio," *European Signal Processing Conference*, 2010.

[10] S. Chen and P. Gopalakrishnan, "Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion," *DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[11] S.S. Chen and P.S. Gopalakrishnan, “Clustering via the Bayesian Information Criterion with Applications in Speech Recognition,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1998, pp. 645-648.

[12] A. Tritschler and R. Gopinath, “Improved Speaker Segmentation and Segments Clustering Using the Bayesian Information Criterion,” *European Conference on Speech Communication and Technology*, Citeseer, 1999, p. 679–682.

[13] M. Cettolo and M. Vescovi, “Efficient Audio Segmentation Algorithms Based on the BIC,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2003.

[14] S. Cheng, H. Wang, and H.-C. Fu, “BIC-based audio segmentation by divide-and-conquer,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 2-5.

[15] S. Pfeiffer, S. Fischer, and W. Effelsberg, “Automatic Audio Content Analysis,” *ACM International Conference on Multimedia*, 1996, pp. 21-30.

[16] T. Zhang and C.-C. Jay Kuo, “Hierarchical System for Content-Based Audio Classification and Retrieval,” *Proceedings of SPIE*, 1998, pp. 398-409.

[17] S.Z. Li, “Content-Based Audio Classification and Retrieval Using the Nearest Feature Line Method,” *IEEE Transactions on Speech and Audio Processing*, vol. 8, 2000, pp. 619-625.

[18] D. Li, I. Sethi, N. Dimitrova, and T. McGee, “Classification of General Audio Data for Content-Based Retrieval,” *Pattern Recognition Letters*, vol. 22, Apr. 2001, pp. 533-544.

[19] G. Guo and S.Z. Li, “Content-based Audio Classification And Retrieval By Support Vector Machines,” *IEEE Transactions on Neural Networks / a Publication of the IEEE Neural Networks Council*, vol. 14, Jan. 2003, pp. 209-15.

[20] P. Wan and L. Lu, “Content-Based Audio Retrieval: A Comparative Study of Various Features and Similarity Measures,” *Proceedings of SPIE*, vol. 6015, 2005, p. 60151H-60151H-8.

[21] M. Baillie and J. Jose, “Audio-Based Event Detection For Sports Video,” *Image and Video Retrieval*, 2003, p. 61–65.

[22] R. Cai, L. Lu, H.J. Zhang, and L.H. Cai, “Highlight Sound Effects Detection in Audio Stream,” *IEEE International Conference on Multimedia & Expo (ICME)*, IEEE, 2003, p. 37–40.

[23] J. Portelo, M. Bugalho, I. Trancoso, J. Neto, A. Abad, and A. Serralheiro, "Non-speech Audio Event Detection," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2009, pp. 1973-1976.

[24] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic Event Detection in Real-Life Recordings," *European Signal Processing Conference*, 2010.

[25] A. Temko, C. Nadeu, and J.I. Biel, "Acoustic Event Detection: SVM-Based System and Evaluation Setup in CLEAR'07," *Multimodal Technologies for Perception of Humans*, 2009, p. 354–363.

[26] M. Kotti, D. Ververidis, G. Evangelopoulos, I. Panagakis, C. Kotropoulos, P. Maragos, and I. Pitas, "Audio-Assisted Movie Dialogue Detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, Nov. 2008, pp. 1618-1627.

[27] L. Lu, H.-jiang Zhang, S. Member, and H. Jiang, "Content Analysis for Audio Classification and Segmentation," *IEEE Transactions on Speech and Audio Processing*, vol. 10, Oct. 2002, pp. 504-516.

[28] R.J.J.H.V. Son, "A Study of Pitch , Formant , and Spectral Estimation Errors Introduced by Three Lossy Speech Compression Algorithms," *Acta Acustica United With Acustica*, vol. 91, 2005, pp. 771 - 778.

[29] M.P. Aylett and S. King, "Single Speaker Segmentation And Inventory Selection Using Dynamic Time Warping Self Organization And Joint Multigram Mapping," *Speech Synthesis Workshop (ISCA)*, Bonn, Germany: Citeseer, 2007.

[30] T. Zhang and C.-C. Jay Kuo, "Audio Content Analysis for Online Audiovisual Data Segmentation and Classification," *IEEE Transactions on Speech and Audio Processing*, vol. 9, May. 2001, pp. 441-457.

[31] H. Sundaram, "Segmentation, Structure Detection and Summarization of Multimedia Sequences," Columbia University, 2002.

[32] E.C. Ozan, S. Tankız, B.O. Acar, and T. Çiloğlu, "Content-Based Audio Event Detection on TV Broadcast," *IEEE Signal Processing and Communications Applications Conference (SIU)*, 2011.

[33] D. Kolossa, "Independent Component Analysis for Environmentally Robust Speech Recognition," Universit\"atsbibliothek, 2007.

[34] H.-gook Kim, N. Moreau, and T. Sikora, *MPEG-7 Audio and Retrieval*, 2006.

[35] B. Liang, H. Yanli, L. Songyang, C. Jianyun, and W. Lingda, "Feature Analysis and Extraction for Audio Automatic Classification," *IEEE International Conference on Systems, Man and Cybernetics*, 2005, pp. 767-772.

[36] "Silence," August, 2010

http://en.wikipedia.org/wiki/Silence.

[37] R.F. Lyon, M. Rehn, S. Bengio, T.C. Walters, and G. Chechik, "Sound Retrieval and Ranking Using Sparse Auditory Representations," *Neural computation*, vol. 22, Sep. 2010, pp. 2390-416.

[38] M. Rehn, R.F. Lyon, S. Bengio, T.C. Walters, and G. Chechik, "Sound Ranking Using Auditory Sparse-Code Representations," *Symposium A Quarterly Journal In Modern Foreign Literatures*, 2009.

[39] E. Akdemir, "Bimodal Automatic Speech Segmentation and Boundary Refinement Techniques," METU, 2010.

[40] A.J. Eronen, V.T. Peltonen, J.T. Tuomi, A.P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, "Audio-based Context Recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, Jan. 2006, pp. 321-329.

[41] S. Petridis and M. Pantic, "Audiovisual Laughter Detection Based on Temporal Features," *International Conference on Multimodal Interfaces, (IMCI)*, 2008, p. 37.

[42] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, "RASTA-PLP Speech AnalysisTechnique," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Ieee, 1992, pp. 121-124 vol.1.

[43] T. Pohle, E. Pampalk, and G. Widmer, "Evaluation of Frequently Used Audio Features for Classification of Music into Perceptual Categories," *International Workshop on Content-Based Multimedia Indexing, (CBMI)*, Citeseer, 2005.

[44] S. Moncrieff, S. Venkatesh, and C. Dorai, "Analysis of Environmental Sounds as Indexical Signs in Film," *Advances in Multimedia Information Processing*, 2001, p. 538–545.

[45] G. Peeters, *A Large Set of Audio Features for Sound Description (Similarity And Classification) in the CUIDADO Project.*, 2004.

[46] I. Mierswa and K. Morik, "Automatic Feature Extraction for Classifying Audio Data," *Machine Learning*, vol. 58, Feb. 2005, pp. 127-149.

[47] P. Herrera, X. Serra, and G. Peeters, “Audio Descriptors and Descriptor Schemes in the Context of MPEG-7,” *International Computer Music Conference*, Citeseer, 1999.

[48] A. Temko, R. Malkin, C. Zieger, D. Macho, C. Nadeu, and M. Omologo, “CLEAR Evaluation of Acoustic Event Detection and Classification Systems,” *Multimodal Technologies for Perception of Humans*, 2007, p. 311–322.

[49] S. Leitich, “Digital Music Libraries,” *Wirtschaftsinformatik*, 2004.

[50] J.A. Bilmes, “A Gentle Tutorial of the EM Algorithm and Its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models,” *International Computer Science Institute*, vol. 4, 1998, p. 126.

[51] S. Ntalampiras, I. Potamitis, and N. Fakotakis, “Exploiting Temporal Feature Integration for Generalized Sound Recognition,” *EURASIP Journal on Advances in Signal Processing*, vol. 2009, 2009, pp. 1-13.

[52] G. Roma, J. Janer, S. Kersten, M. Schirosa, P. Herrera, and X. Serra, “Ecological Acoustics Perspective for Content-Based Retrieval of Environmental Sounds,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2010, 2010, pp. 1-11.

[53] A. Lima, H. Zen, Y. Nankaku, C. Miyajima, K. Tokuda, and T. Kitamura, “On the Use of Kernel PCA for Feature Extraction in Speech Recognition,” *European Conference on Speech Communication and Technology*, 2003, pp. 2625-2628.

[54] C. Shang and D. Barnes, “Combining Support Vector Machines and Information Gain Ranking for Classification of Mars McMurdo Panorama Images,” *IEEE International Conference on Image Processing (ICIP)*, IEEE, 2010, p. 1061–1064.

[55] L. Barrington, A. Chan, D. Turnbull, and G. Lanckriet, “Audio Information Retrieval Using Semantic Similarity,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Ieee, 2007, p. II–725.

[56] R. Setiono, “Chi2: Feature Selection and Discretization of Numeric Attributes,” *IEEE International Conference on Tools with Artificial Intelligence*, IEEE Comput. Soc. Press, , pp. 388-391.

[57] X. Jin, A. Xu, R. Bie, and P. Guo, “Machine Learning Techniques and Chi-Square Feature Selection for Cancer Classification Using SAGE Gene Expression Profiles,” *Data Mining for Biomedical Applications*, 2006, p. 106–115.

[58] C.D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[59] W.-H. Cheng, W.-T. Chu, and J.-L. Wu, "Semantic Context Detection Based on Hierarchical Audio Models," *International Workshop on Multimedia information retrieval (MIR)*, 2003, p. 109.

[60] C. Clavel, T. Ehrette, and G. Richard, "Events Detection for an Audio-based Surveillance system," *IEEE International Conference on Multimedia & Expo (ICME)*, IEEE, 2005, p. 1306–1309.

[61] J.G. Dy and C.E. Brodley, "Feature Selection for Unsupervised Learning," *The Journal of Machine Learning Research*, vol. 5, 2004, p. 845–889.

[62] J.-ching Wang, J.-fa Wang, K.W. He, and C.-shu Hsu, "Environmental Sound Classification using Hybrid SVM/KNN Classifier and MPEG-7 Audio Low-Level Descriptor," *IEEE International Joint Conference on Neural Network Proceedings*, 2006, pp. 1731-1735.

[63] A. Öztürk, "SVM Classification for Imbalanced Datasets with Multi Objective Optimization Framework," Koc University, 2009.

[64] M. Pirooznia, J.Y. Yang, M.Q. Yang, and Y. Deng, "A Comparative Study of Different Machine Learning Methods on Microarray Gene Expression Data," *BMC genomics*, vol. 9 Suppl 1, Jan. 2008, p. S13.

[65] W.-ta Chu, "Generative and Discriminative Modeling toward Semantic Context Detection in Audio Tracks," *International Multimedia Modelling Conference*, 2005, pp. 38-45.

[66] C.J.C. Burges, "A Tutorial On Support Vector Machines For Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, 1998, p. 121–167.

[67] Y.C. Chien-Chang Lin, Shi-Huang Chen, Trieu-Kien Truong, "Audio Classif cation and Categorization Based on Wavelets and Support Vector Machine," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 13, 2005.

[68] E. Doğan, "Content-Based Audio Management and Retrieval Systems for News Broadcasts," METU, 2009.

[69] P. Boersma and D. Weenink, "Praat: Doing phonetics by computer," 2005.