USC-SIPI REPORT #430

Classification and Retrieval of Environmental Sounds

by

Sachin Chachada

May 2016

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Viterbi School of Engineering
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Suite 400
Los Angeles, CA 90089-2564 U.S.A.

CLASSIFICATION AND RETRIEVAL OF ENVIRONMENTAL SOUNDS

by

Sachin Chachada

A Dissertation Presented to the

FACULTY OF THE GRADUATE SCHOOL

UNIVERSITY OF SOUTHERN CALIFORNIA

In Partial Fulfillment of the

Requirements for the Degree

DOCTOR OF PHILOSOPHY

(ELECTRICAL ENGINEERING)

January 2016

This work is dedicated to my loving family!

Acknowledgments

First of all, I would like to thank Dr. C.-C. Jay Kuo for not only being an advisor for research, but also being a life coach. He has fostered an intellectual, nurturing and familial environment at Media Communications Lab. I am grateful to be a part of MCL family and, hence, would like to thank all its members. In particular, I would like to thank Hang Yuan, Xiaqing Pan, Martin Gawecki, Sanjay Purushotham, Chun-Ting Haung, Jian Li, Xiang Fu and Chen Chen for all the insightful theoretical discussions, socio-political rants and heartwarming company which got me through the ebb and flow of life as a Ph.D.student.

I would also like to thank Dr. Panos Georgiou, Dr. Bartlett Mel, Dr. Shrikanth Narayanan and Dr. Keith Jenkins for their valuable suggestions and ideas that helped in perfecting this dissertation.

Contents

List of Figures

List of Tables

Abstract

Audio processing research has primarily been focused on speech and music signals. However, research on general audio or environmental sound processing, i.e. audio signals other than speech an music, has been scant. Owing to its numerous applications such as those in the fields of surveillance, bio-diverity monitoring, robot audition, etc., there is a need of a good Environmental Sound Recognition (ESR) system. Hence, in this dissertation, we work on classification and retrieval systems for environmental sounds.

In order to build an efficient ESR model, it is important to be able to characterize environmental sounds. Environmental sounds are rich in both context and content. What sets them apart from music and speech is their non-stationary nature. Hence, recent work has focused on the study and development of features for environmental sound that focus on their non-stationary characteristics. This work first focuses on assessing these features by providing a common test database. Analysis of these features helped us in understand their power and limitations. Features motivated by dual time-frequency (TF) representation have become quite popular and have been proven successful. Among these, sparse representation over Gabor dictionary is a popular and recent feature. However, we will show that these features fail when applied to a large and diverse database. Thus, we propose a modification to these features by first filtering a signal using a narrow-band filterbank and then extracting these features for each filtered

band-limited signal. The proposed features, Narrow Band Time Frequency features, are shown to be robust for large scale databases.

Environmental sounds are complex signals. Hence, we believe it is hard to find a single feature which would work for any database, and would be scalable to a large number of sounds. Thus, we leverage the machine learning approaches of decision fusion, also known as Ensemble learning, classifier fusion, multiple-experts, and propose a multi-classifier system for this task. The proposed Para-Boost Multi-Classifier (PB-MCS) model, takes the advantages of all the features and improves the overall performance of ESR system. PB-MCS uses vertical decomposition, i.e. decomposition of data-matrix along feature dimension, to form individual experts and finally combine the predictions of these experts. We also propose several variations of PB and study them in detail.

Considering the exponential growing environmental sound data on the Internet, we need a good content based retrieval system. Audio data on the Internet is often tagged with class labels and hence it is reasonable to assume that the data-base is partially labeled. By allowing database to be partially labeled, we take the advantage of labels to narrow the search for relevant sounds and allow room for growing the database without assigning labels to each document. To this end, we present a two stage content based (query-by-example) environmental sound retrieval system. In Stage I, we first exploit the signal characteristics such as time localized and frequency localized energy distribution to do a broad categorization of environmental sounds. This not only reduces the potential query matching complexity, but also enables us to customize ensuing steps that exploit these characteristics of environmental sounds. Next, for each category, a classifier is trained to predict labels for unlabeled data in the database and also narrow search range for a query by assigning it multiple, yet limited, class labels. In Stage II, we propose a novel feature and a scoring scheme to do local matching and ranking. First, each audio clip is segmented based on any extracted features. This segmentation is done using Mean

Shift approach, and hence is an unsupervised segmentation. Then, relevant segments are extracted for each clip, and each segment is represented by its point of convergence in the feature space. The audio signal is finally represented by its energy distribution over each segment thereby capturing the temporal variations of the audio signal in feature space. Given a query, first audio segments of a document are mapped to those of the query. Then the document is assigned a score based on energy distribution of mapped segments only.

Chapter 1

Introduction

1.1 Significance of the Research

A considerable amount of research has been made towards modeling and recognition of environmental sounds over the past decade. By environmental sounds, we refer to various quotidian sounds, both natural and artificial (*i.e.* sounds one encounters in daily life other than speech and music). In today's media driven world, Environmental sound recognition (ESR) finds applications in many fields such as efficient audio search and retrieval [53, 15],robot navigation[9, 61], assisted living for elderly people[7, 49], smart home[54], surveillance [12, 42] and bio-acoustic applications[3, 57]. Simply put, ESR plays a pivotal part in recent efforts to perfect machine audition.

Among various types of audio signals, speech and music are two categories that have been extensively studied. In its infancy, ESR algorithms were a mere reflection of speech and music recognition paradigms. However, on account of considerably non-stationary characteristics of environmental sounds, these algorithms proved to be ineffective for large-scale databases. For example, the speech recognition task often exploits the phonetic structure that can be viewed as a basic building block of speech. It allows us to model complicated spoken words by breaking them down into elementary phonemes that can be modeled by the Hidden Markov Model (HMM) [38]. In contrast, general environmental sounds, such as that of a thunder or a storm, do not have any apparent sub-structures like phonemes. Even if we were able to identify and learn a dictionary of basic *units* (analogous to phonemes in speech) of these events, it would be difficult

to model their variation in time with HMM as their temporal occurrences would be more random as against preordained sequence of phonemes in speech. Similarly, as compared to music signals, environmental sounds do not exhibit meaningful stationary patterns such as melody and rhythm [41]. To the best of our knowledge, there was only one survey article on the comparison of various ESR techniques done by Cowling and Sitte [11] about a decade ago.

Research on ESR has significantly increased in the last decade. Recent work has focused on the appraisal of non-stationary aspects of environmental sounds, and several new features predicated on non-stationary characteristics have been proposed. These features, in essence, strive to maximize their information content pertaining to signal's temporal and spectral characteristics as bounded by the uncertainty principle. For most real life sounds, even these features exhibit non-stationarity when observed over a long period of time. To capture these long-term variations, sequential learning methods have been applied.

It becomes evident that ESR methods not only have to model non-stationary characteristics of sounds, but also have to be scalable and robust as there are numerous categories of environmental sounds in real life situations. Despite increased interest in the field, there is no single consolidated database for ESR, which often hinders benchmarking of these new algorithms.

All these challenges motivate my research - a quest for a good ESR system.

1.2 Contributions of the Research

In this work, we present a new feature to characterize environmental sounds. Motivated by multiple classifier systems, we also present several novel models for ESR. Finally, we proposed an effective content based framework for retrieval.

- Narrow Band Time Frequency (NBTF) Features: Time-frequency representations are powerful tools to understand stochastic signals. However, several TF features proposed in the recent years fail to effectively utilize this powerful tool. In particular, sparse representation over Gabor dictionary have shown impressive results in other applications. However, features proposed using this approach proved inadequate for a large scale ESR database. Thus, we propose NBTF features which characterize a signal using TF representation of band limited filters thereby making them robust for large scale databases.

- Para-Boost Multiple Classifier System (PB-MCS): Ensemble learning, classifier fusion, multiple-experts - whatever you call them, are known for their efficaciousness. It only seems natural to gather opinions of multiple experts when making a complex decision. We propose a novel fusion approach, PB-MCS, which utilize the concepts of two very different ensemble approaches - random subspace and stacked generalization. Though both are ensemble systems, the system of belief each follows is very different. We combine their principle ideas to form a new system which exploits both the ideas of diverse classification systems and diverse projections of data itself.

- Content Based Environmental Sound Retrieval: In this work, we present a two stage content based environmental sound retrieval system. This query-by-example retrieval system assumes that the database is partially labeled. In Stage I, we do a broad categorization of environmental sounds into three categories, namely Time Localized Signals, Frequency Localized Signals, and Others. This signal-characteristic based categorization enables us to customize ensuing processing steps for retrieval. For each category, a classifier is trained to predict labels for unlabeled data in the database. This classifier would also be used to narrow down

search space for a query. In Stage II, we propose a novel feature and a scoring scheme to do local matching and ranking. First, an audio signal is segmented in an unsupervised manner using Mean Shift algorithm. The segmentation is done on features extracted from the audio instead of using the raw audio itself. Each segment is represented by its point of convergence in the feature space. The audio signal is finally represented by its energy distribution over each segment. This representation is content based and is done independently for each document on the database and for the query. Given a query, first audio segments of a document are mapped to those of the query. Then the document is assigned a score based on energy distribution of mapped segments only.

1.3 Organization of the Thesis

The rest of the proposal is organized as follows: Chapter 2 reviews state-of-the art approaches in depth. In Chapter 3, we introduce NBTF, and do a thorough evaluation of these features as compared to benchmark approaches on our own database. PB-MCS and its variants are presented in Chapter 4. In Chapter 5, we propose and discuss Content based Environmental Sound Retrieval system. Finally in Chapter 6, concluding remarks and future work are presented.

Chapter 2

Background

2.1 Environmental Sound Processing Schemes

Before delving into the details of various ESR techniques, we will first describe three commonly used environmental sound processing schemes in this section.

Framing-based processing

Audio signals to be classified are first divided into frames, often using a Hanning or a Hamming window. Features are extracted from each frame and this set of features is used as one instance of training or testing. A classification decision is made for each frame and, hence, consecutive frames may belong to different classes. A major drawback of this processing scheme is that there is no way of selecting an optimal framing-window length suited for all classes. Some sound events are short-lived (*e.g.* gun-shot) as compared to other longer events (*e.g.* thunder). If the window length is too small, the long-term variations in the signal would not be well captured by the extracted features, and the framing method might chop events into multiple frames. On the other hand, if the window length is too large, it becomes difficult to locate segmental boundaries between consecutive events and there might be multiple sound events in a single frame. Also, one has to rely on features to extract non-stationary attributes of the signal since such a model does not allow the use of sequential learning methods.

Sub-framing-based processing

Each frame is further segmented into smaller sub-frames, usually with overlap, and features are extracted from each sub-frame. In order to learn a classifier, features extracted from sub-frames are either concatenated to form a large feature vector or averaged so as to represent a single frame. Another possibility is to learn a classifier for each sub-frame and make a collective decision for the frame based on class labels of all sub-frames (*e.g.*, a majority voting rule). This model allows the use of both non-stationary features and sequential classifiers. Even with a non-sequential classifier, this processing scheme can represent each frame better as the collective distribution over all sub-frames allows one to model intra-frame characteristics with greater accuracy. This method offers more flexibility in segmenting consecutive sound events based on class labels of sub-frames.

Sequential processing

Audio signals are still divided into smaller units (called a segment), which is typically of 20-30 ms long with 50% overlap. The classifier makes decisions on class labels and segmentation both based on features extracted from these segments. As compared to the above two methods, this method is unique in its objective to capture the inter-segment correlation and the long-term variations of the underlying environment sound. This can be achieved using a sequential signal model such as the Hidden Markov Models (HMM).

Any ESR algorithm basically follows one of the above three processing schemes with minor variations in its preprocessing and feature selection/reduction schemes. For example, a pre-emphasis filter can be used to boost the high frequency content or an A-weight filter can be used for equalized loudness. For feature selection/reduction, there is an arsenal of tools to choose from [29, 36, 52]. We will not pay attention to these minor differences in later sections.

2.2 Features for Environmental Sound Recognition

Features commonly used for ESR can be broadly classified into different categories based on what kind of information they encode. In a manner of speaking, each feature tries to capture some aspect of audio signal based on certain assumptions. We will introduce these features categorized on nature of these characteristics and assumptions.

2.2.1 Stationary ESR Techniques

Features developed for speech/music based applications have been traditionally used in stationary ESR techniques. These features are often based on psychoacoustic properties of sounds such as loudness, pitch, timbre, etc. A detailed description of features used in audio processing was given in [32], where a novel taxonomy based on the properties of audio features was provided (see Fig. 2.1).

Features such as Zero-Crossing Rate (ZCR), Short-Time Energy (STE), Sub-band Energy Ratio, Spectral Flux, etc. are easy to compute and used frequently along with other refined set of features. These features provide rough measures about temporal and spectral properties of an audio signal. For more details on basic features, we refer to [13, 32, 35, 37].

Cepstral features are widely used features. They include: Mel-Frequency Cepstral Coefficients (MFCC) and their first and second derivatives (ΔMFCC and $\Delta\Delta$MFCC), Homomorphic Cepstral Coefficients (HCC), Bark-Frequency Cepstral Coefficients (BFCC), etc. MFCC were developed to resemble the human auditory system and have been successfully used in speech and music applications. As mentioned before, due to lack of a standard ESR database, MFCC are often used by researchers for benchmarking their work. A common practice is to concatenate MFCC features with newly developed features to enhance the performance of a system.

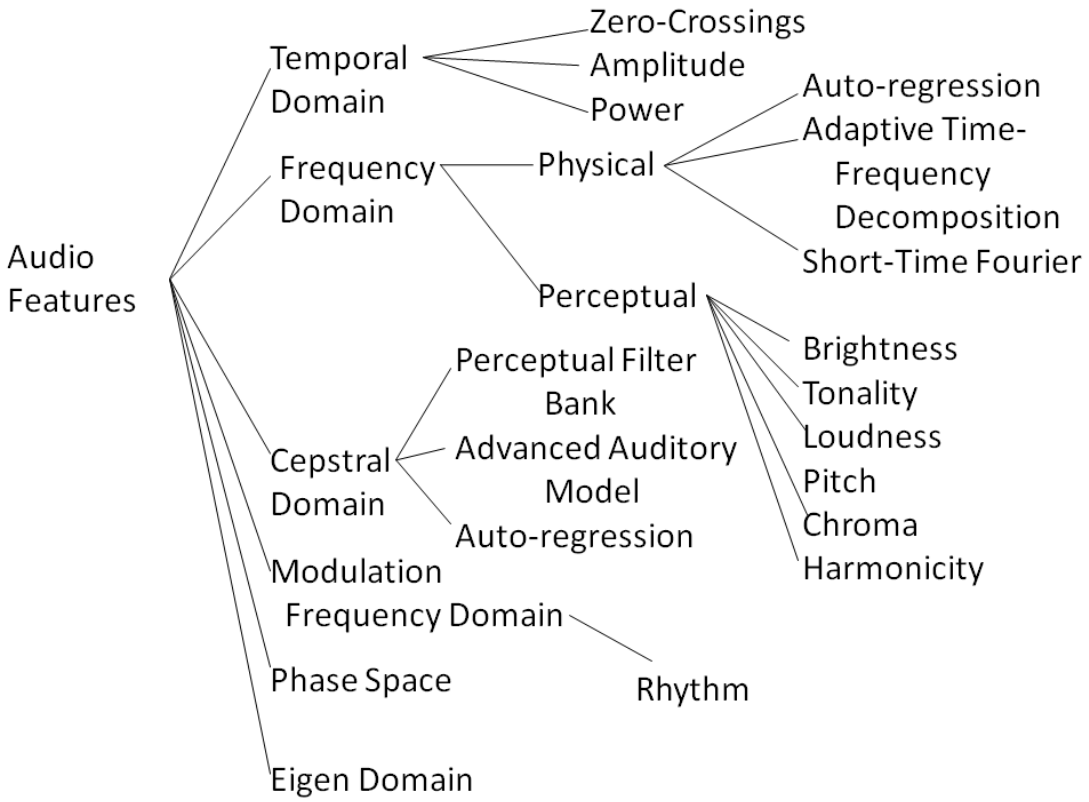

Figure 2.1: Taxonomy for audio features as proposed in [32].

MPEG-7 based features are also popular for speech and music applications. They demand low computational complexity and encompass psychoacoustic (or perceptual-based) audio properties. Wang *et al.* [56] proposed to use low-level audio descriptors such as Audio Spectrum Centroid and Audio Spectrum Flatness with a hybrid classifier constituted of Support Vector Machine (SVM) and K-Nearest Neighbors (KNN). They converted the classifier outputs from SVM and KNN into probabilistic scores and fused them to improve classification accuracy. Muhammad *et al.* [33] combined several low-level MPEG-7 descriptors and MFCC and used Fisher's Discriminant Ratio (F-Ratio) to discard irrelevant features. Although MPEG-7 features perform better than MFCC,

MFCC and MPEG-7 descriptors are shown to be complementary to each other and, when used together, the classification accuracy can be improved.

Auto-regression based features, in particular, Linear Prediction Coefficients (LPC), have been prevalent in speech processing applications. Linear Prediction Cepstrum Coefficients (LPCC), which are an alternate representation of LPC, are also commonly used. However, LPC and LPCC embody the source-filter model for speech and, hence, they are not useful for ESR. Tsau *et al.* [47] proposed the use of the Code Excited Linear Prediction (CELP) based features along with the LPC, pitch and pitch gain features. Since CELP uses a fixed codebook for excitation of a source-filter model, it is more robust than LPC. Tsau *et al.* [47] reported improved performance over MFCC. CELP and MFCC together further increase the classification accuracy, specially noticeable for classes like rain, stream and thunder which are difficult to recognize.

ESR algorithms relying on the sub-framing processing scheme usually learn signal-models in each sub-frame and, thus, do not utilize the temporal structure. One variation to exploit the temporal structure is when a signal-model is learned based on features from all ordered sub-frames such as HMM. Another example was recently proposed by Karbasi *et al.* [23], which attempted to capture the temporal variation among sub-frames in a new set of features called "Spectral Dynamic Features (SDF)" as detailed below.

Let $x_{sb}(i)$ denote the i^{th} sub-frame, with $i \in [1, N]$. From each sub-frame $x_{sb}(i)$, MFCC and other features are extracted in a vector y_i with dimension $L \times 1$. Let $Y = [y_1, \ldots, y_N]$ be a matrix with columns y_i of feature vectors for N sub-frames. For each row of Y, the N-point FFT is applied followed by the logarithmic filter bank, and then followed by the N-point DCT to yield the final set of features. This method essentially extracts cepstral features (MFCC-like features) considering each row of Y as a time series. For example, if 13 MFCC coefficients are extracted from each sub-frame, then we end up with 13 time series, one for each dimension. The cepstral features are

evaluated for each of these time series by capturing the dynamic variation of sub-frame features over the entire frame. The superior performance of SDF against several conventional features such as ZCR, LPC, MFCC under three classifiers (*i.e.*, KNN, GMM and SVM) was demonstrated. It was shown in [23] that the combined features of MFCC and ΔMFCC give the performance bound of *static* features, which is not improved by adding more conventional features. A system with a feature vector consisting of ZCR, Band-Energy, LPC, LPCC, MFCC and ΔMFCC, performs poorly as compared to that with only MFCC and ΔMFCC under the SVM or GMM classifiers. In contrast, the *dynamic* feature set, SDF, achieves an improvement of $10 - 15\%$ over the *static* bound.

Filter-banks are often used to extract features local to smaller bands, encapsulating spectral properties effectively. On the other hand, the auto-correlation function (ACF) represents the time-evolution and has an intimate relationship with the power spectral density (PSD) of the underlying signal. Valero and Alias [50] proposed a new set of features called the Narrow-Band Auto Correlation Function features (NB-ACF). The extraction of NB-ACF features can be explained using Fig. 2.2. First, a signal is passed through a filter bank with $N = 48$ bands whose center frequencies being tuned to the Mel-scale. Then, the sample ACF of the filtered signal in the ith band is calculated, which is denoted by $\Phi_i(\tau)$. One can calculate four NB-ACF features based on each ACF as follows.

1. $\Phi_i(0)$: Energy at lag $\tau = 0$. It is a measure of the perceived sound pressure at the i^{th} band.

2. τ_{i_1}: Delay of the first positive peak which represents the dominant frequency in the i^{th} band.

3. $\Phi_{i_1}(\tau_{i_1})$: Normalized ACF of the first positive peak. It is related to the periodicity of the signal and, hence, gives a sense of pitch of the filtered signal at the i^{th} band.

4. τ_{i_e}: Effective duration of the envelope of normalized ACF. It is defined as the time taken by normalized ACF to decay 10 dB from its maximum value, and it is a measure of reverberation of the filtered signal at the i^{th} band.

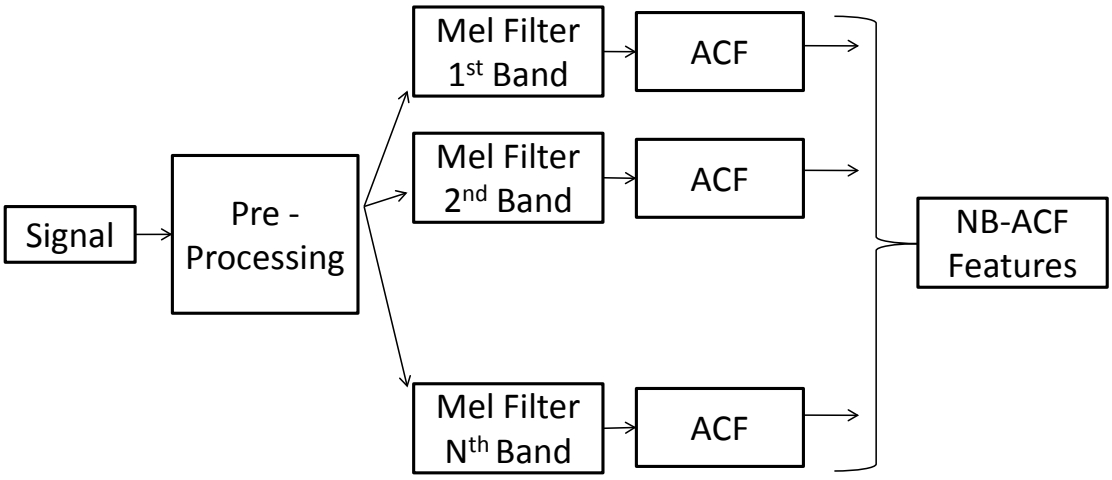

Figure 2.2: Illustration of the NB-ACF feature extraction process.

As a rule of thumb, the sample ACF is meaningful up to lag τ_{opt} if and only if the signal length is at least four times the lag length. This demands a sub-frame length to be much larger than that used in sub-framing processing. It is recommended in [50] that a be sub-frame of size 500ms with an overlap of 400 ms in a frame of 4 seconds. Finally, KNN and SVM classifiers are used for decision making in each sub-frame. The performance of NB-ACF features was compared with MFCC and Discrete Wavelet Transform (DWT) coefficients with a data-set consisting of 15 environmental scenes. Dynamically changing scenes, such as *office*, *library* and *classroom*, pronouncedly benefited from the new NB-ACF features. It is well known that ACF is instrumental in the design of linear predictors for time-series because they capture the temporal similarity/dissimilarity well. As a result, the NB-ACF features offer better performance for wide-sense-stationary (WSS) signals than most static features discussed in this section.

2.2.2 Non-stationary ESR Techniques

Any signal can be analyzed in both time and frequency domain. Both these signal representations provide different perspectives of a signal from a physical standpoint. Time information gives exact measurable representation of signal, be it vibrations as in case of audio, or light and color intensities in case of images, and so on. On the other hand, Frequency domain methods such as Fourier Transform give an idea of average power over various constituent frequencies of the signal, thereby describing the nature of the physical phenomenon constituting the signal. However, the analysis tools when restricted to only one domain, take measurements with the assumption that the progenitorial phenomena responsible for signal production do not vary with time. Thus, features extracted using these tools work well when this assumption of "stationarity" is satisfied. However, as discussed before, real life audio signals often violate this assumption. Such signals are assumed to have time varying characteristics, and thus such signals can be described as "non-stationary". A class of tools, known as time-frequency analysis methods, are employed when dealing with such signals. In the following sections, we will discuss features derived from such time-frequency analysis tools.

Wavelet-Based Methods

For decades, Wavelet Transforms have been used to represent non-stationary signals since they offer representation in both time and frequency space. As compared to Fourier Transform, which uses analytic waves to decompose a signal, Wavelet Transforms use "wavelets" which are nothing but "short waves" with finite energy[10]. Time-varying frequency analysis of a signal is made possible by the finite-energy property of these wavelets. An analytic function, $\Psi(t)$ must satisfy following conditions

- A wavelet must have finite energy

$$\int_{-\infty}^{+\infty} |\Psi(t)|^2 dt < \infty \tag{2.1}$$

- The admissibility condition [20] must be met by the Fourier Transform of the wavelet, $\mathbf{\Psi}(\omega)$

$$\int_{0}^{\infty} \frac{|\mathbf{\Psi}(\omega)|^2}{\omega} df < \infty \tag{2.2}$$

Such an analytic function is admissible as a "mother wavelet" and can be used to generate daughter wavelets by scaling and shifting, thereby enabling more accurate localization of signal in time-frequency space. Together these functions can be used to decompose a given signal to give time-varying frequency profile.

The performance of commonly employed features for audio recognition, including Mel-Frequency Cepstral Coefficients (MFCC), Homomorphic Cepstral Coefficients (HCC), time-frequency features derived using Short-Term Fourier Transform (STFT), Discrete Wavelet Transform (DWT) and Continuous Wavelet Transform (CWT) was compared by Cowling and Sitte in [11], where the Learning Vector Quantization (LVQ), Artificial Neural Networks, Dynamic Time Warping (DTW) and Gaussian Mixture Models (GMM) were used as classifiers. The experiments were conducted on three types of data – speech, music and environmental sounds. For the environmental sound, the data set consisted of 8 classes, and the framing-based processing scheme was adopted. It was reported that the best performance for ESR was achieved with CWT features with the DTW classifier, which was comparable to that of MFCC features with the DTW classifier. It is surprising that CWT, which is a time-frequency representation, and MFCC gave very similar results while DWT and STFT did not give good performance. It was noted in [11] that the dataset was too small to make any meaningful comparison between MFCC and CWT. Given other factors being equal, MFCC features can be more

favored than CWT features because of their lower computational complexity. DTW was clearly the best classifier in the test, yet the claim should be further verified by a larger environmental sound database.

Han and Hwang [22] used the Discrete Chirplet Transform (DChT) and the Discrete Curvelet Transform (DCuT) along with several other common features such as MFCC, ZCR, etc. When compared, all features gave similar performance, yet significant improvement was observed when they were used together.

Valero and Alias [51] adapted the Gammatone mother function to meet wavelet admissibility conditions, used the squared sum of Gammatone representations of signal as features, called the Gammatone wavelet features, and adopted the SVM classifier. A comparable performance was observed between Gammatone wavelet features and DWT. When both features were used together, classification accuracy was improved even in noisy conditions. Gammatone features perform well in classes such as footsteps and gunshots due to their capability in characterizing transient sounds.

Umapathy *et al.* [48] proposed a new set of features based on the binary wavelet packet tree (WPT) decomposition. More recently, Su *et al.* [46] used a similar approach to recognize sound events in an environmental scene consisting of many sound events. This ESR algorithm was conducted with the framing-based processing scheme. The signal in the ith frame, x_i, is first transformed to a binary WPT representation denoted by $\Omega_{j,k}$, where j is the depth of the tree and k is the node index at level j. Each subspace $\Omega_{j,k}$ is spanned by a set of basis vectors $\{\mathbf{w}_{j,k,l}\}_{l=0}^{2^u-1}$, where 2^u is the length of x_i. Then, we have

$$x_i = \sum_{j,k,l} [\alpha_{j,k,l}]_i \ \ \mathbf{w}_{j,k,l}, \tag{2.3}$$

where $\alpha_{j,k,l}$ is the projection coefficient at node (j,k). Once all training samples are decomposed to a binary WPT, the Local Discriminant Bases (LDB) algorithm is used to identify the most discriminant nodes of the WPT. The LDB algorithm can be simply

described below. For each pair of classes in the data set, one can determine a set of Q discriminatory nodes based on a dissimilarity measure. Two dissimilarity measures were proposed in [46]:

1. the difference of normalized energy

$$D_1 = E_1^{(j,k)} - E_2^{(j,k)}$$

   of the two sound classes at the same node (j, k);

2. the ratio of the variances of projection coefficients of the two sound classes at node (j, k),

$$D_2 = var[\mathbf{v}_1^{(j,k)}]/var[\mathbf{v}_2^{(j,k)}],$$

   where $\mathbf{v}_i^{(j,k)}$ is the vector of variance of locally grouped coefficients at node (j, k).

Strictly speaking, none of these two dissimilarity measures are distance metrics. The selected Q nodes should be consistent. It was recommended to conduct multiple trials with randomly selected training samples from two classes, and consistent nodes should be selected from these random trials. The above process should be repeated among all possible class pairs. Finally, we select H nodes that occur most frequently among the Q nodes for each pair, and use coefficients and/or dissimilarity measure quantities at these H nodes as features.

The LDA-based classifier was used in [48] while the KNN and HMM were used in [46]. It was observed in [48] that WPT-LDB and MFCC features gave similar performance, yet much better performance was achieved when the two were combined together. It was reported in [46] that MFCC performed better than WPT-LDA, and a significant improvement could be obtained by combining the two features. Note that

the classification performance in [46] was given for environmental scenes rather than individual events.

Despite being time-frequency features, the performance of wavelet features is not better than that of MFCC features but at a comparable level. When being combined with MFCC, the performance does improve yet the required complexity overhead to extract wavelet features might not always justify the gain in classification accuracy except for the Gammatone features. The Gammatone features are proved to be complementary to MFCC owing to their strong capability in representing impulsive signal classes such as footsteps and gun-shots.

Sparse-Representation-Based Methods

Chu *et al.* [8] proposed to use the Matching Pursuit (MP) based features for ESR. The basis MP (BMP) is a greedy algorithm used to obtain a sparse representation of signals based on atoms in an over-complete dictionary. Given signal x and an over-complete dictionary $D = [d_1, d_2, ...]$, BMP obtains the sparse representation of x on D as follows.

1. Initialize the residue at the 0^{th} iteration as $R^0 x = x$
2. For $t = 1$ to T
   (a) Select the atom with the largest inner product with the residue via
   $$d_t = \max_i < R^{t-1}x, d_i > .$$
   (b) Update the residue via
   $$R^t x = R^{t-1}x - \alpha_t d_t,$$
   where $\alpha_t =< R^{t-1}x, d_t >$ is the projection coefficient of $R^{t-1}x$ on d_t.

3. The BMP projection of x on D is given by

$$\hat{x} = \sum_{i=1}^{T} \alpha_i d_i$$

One stopping criterion for this algorithm is a fixed number of iterations (atoms), T. Another one is to use the energy of the residual signal, i.e. decomposition stops at t when $||R^{t-1}x||^2 <$ Threshold.

An over-complete Gabor dictionary consisting of frequency modulated Gaussian functions (called Gabor atoms) was used in [8]:

$$g_{s,u,\omega,\theta} = \frac{K_{s,u,\omega,\theta}}{\sqrt{s}} e^{-\pi(n-u)^2/s^2} \cos\left[2\pi\omega(n-u)+\theta\right], \tag{2.4}$$

where s, u, ω and θ are atom's scale, location, frequency and phase, respectively, and $K_{s,u,\omega,\theta}$ is a normalization constant so that $||g_{s,u,\omega,\theta}||^2 = 1$.

The following parameters were chosen: $s = 2^p (1 \leq p \leq 8)$, $u = \{0, 64, 128, 192\}$, $\omega = 0.5 \times 35^{-}2.6\ i^{2.6}\ (0 \leq i \leq 35)$, and $\theta = 0$, with each atom of size $N = 256$ given signal sub-frames of size 256 at a sampling frequency of 22.05 kHz. The classification accuracy is not affected much for $T > 5$ so that the first $T = 5$ atoms in the MP algorithm is used. The selected features are the mean and the variance of scale and frequency parameters of the 5 selected atoms, *i.e.*, $[\mu_s, \mu_w, \sigma_s, \sigma_w]$, which are referred to as the MP-Gabor features. The location and phase parameters are ignored. It adopts the sub-framing processing scheme with a frame of 4 seconds and a sub-frame of 0.11 ms with 50% overlap. For classification, KNN and GMM classifiers were tested. The MP-Gabor features perform marginally better than MFCC, and the classification accuracy is further improved when used together with MFCC. Sound classes with broad-spectrum fare well with the MP-Gabor features, but classes with highly non-stationary characteristics such as thunder sounds have poorer recognition accuracy.

To improve the performance of MP-Gabor features, Sivasankaran and Prabhu [43] proposed several modifications. First, they construct a signal-dependent over-complete dictionary (rather than using a fixed dictionary) for signals. The normalized frequency scale is divided into N sub-bands, and the normalized energy present in each sub-band is calculated using DFT coefficients. Suppose that a total of N_f frequency points are to be used in the dictionary. The number of frequency points in each sub-band is proportional to its normalized energy and equally-spaced frequency points in each sub-band are used. Second, the Orthogonal Matching Pursuit (OMP), which is a variant of BMP, was used. At each iteration, OMP computes the orthogonal projection matrix using previously selected atoms and calculates projection coefficients using this projection matrix. Third, the weighted sample mean and variance are used. They achieved high classification accuracy by using modified MP-Gabor features and MFCC yet without performance benchmarking with other methods. The modified MP-Gabor and MFCC features together perform well for most sound classes including thunder. The only two classes with lower classification accuracy were ocean and rain. They are actually quite similar when heard for a small duration of time.

In both the works discussed above, features are derived only from scale and frequency parameters, and the sparse representation coefficients themselves are ignored. This makes sense when the number of atoms selected is very small ($N = 5$ for both works). Considering that the coefficients can take on any real value, only few of such values if used would be a noisy representation of a class. However, if we were to do a very accurate decomposition of a signal by using large number of atoms, then with sufficient samples from one class, one can hope to use the coefficient information to represent a class more accurately. In a very recent work which is inspired by this principle, Wang *et al.* [55] proposed to represent signals as a 2-D map in scale and frequency parameter space of sparse decomposition using a large number of atoms ($N = 60$).

For 16 kHz sampling frequency, authors propose 8 non-uniform frequency parameters, $\omega = 2\pi f, f \in \{150, 450, 840, 1370, 2150, 3400, 5800\}$ Hz, which were based on psychoacoustic studies of human auditory system. The remaining Gabor dictionary parameters were same as those in [8]. Once the decomposition is done, sub-frame of a signal is represented as a matrix with sparse representation coefficients in cells corresponding to atom's scale and frequency parameters. Finally, mean of 16 contiguous sub-frame matrices are averaged to have a stable representation for a single frame. To reduce computational complexity, and have better representation capabilities within single class, PCA and LDA are used to extract final features - Non-uniform Map features. Finally, SVM is used for classification. The authors test the proposed features on 17 classes, and compare their performance with MP-Gabor+MFCC features. They show an average improvement of about 3%. Before wrapping up the discussion on this work, we would like to point out that features proposed in both [8] and [43] can be seen as special cases of Non-uniform Map features. In order to reduce dimensionality of Non-uniform Map, instead of using PCA followed by LDA, if we simply keep top 5 most atoms, and give them equal weight by setting their coefficients to one, we obtain features akin to MP-Gabor features, i.e. both have same information content, just the representation is slightly different.

Yamakawa *et al.* [60] compared the Haar, Fourier and Gabor bases with the HMM classifier using the sequential processing scheme. Instead of using the mean and the standard deviation of scale and frequency parameters of MP-Gabor atoms, they concatenated them to construct a feature vector. Since MP is a greedy algorithm, one may not expect ordered atoms to offer an accurate approximation to non-stationary signals. Due to the use of the HMM classifier, results for Gabor features are still good when the sound classes were restricted to impulsive sounds. The classification accuracy of Haar wavelets was low in the experiment, which is counter-intuitive since the Haar basis

matches the impulse-like structure well in the time domain. This work does show that HMM can better capture the variations in features when 6 mixtures are used in GMM to model hidden states. Also, the performance of time-frequency Gabor features and stationary Fourier features are comparable.

To conclude, MP-based features that are capable of extracting the information of high time-frequency resolution improve the performance of an ESR system when used together with the popular MFCC. Moreover, classification accuracy can be further improved using sequential learning methods such as HMM.

Power-Spectrum-Based Methods

The spectrogram provides useful information about signal's energy in a well localized time and frequency region. It is an intuitive tool to extract transient and variational characteristics of environmental sounds. However, it is not easy to use the spectrogram features in learning models for ESR for a small database due to its higher dimensionality.

Khunarsal *et al.* [24] used the sub-framing processing scheme to calculate the spectrogram as the concatenation of the Fourier Spectrum of sub-frames and adopted the Feed-Forward Neural Network(FFNN) and KNN for classification. Extensive study was done on the selection of spectrogram size parameters, the audio signal length, the sampling rate and other model parameters needed for accurate classification. The features were compared with MFCC and LPC and MP-Gabor features. The spectrogram features perform consistently better against MFCC and LPC and give comparable results against MP-Gabor features. Although a combination of the spectrogram, LPC and MP-Gabor features gives the best results, classification results with other feature combination are comparable to the best one. This implies that there is redundancy in these features.

Recently, Ghoraani and Krishnan [17] proposed a novel feature extraction method based on the spectrogram using the framing processing scheme. First, the MP representation for a signal is achieved with the Gabor dictionary that has fine granularity in scale, frequency, position and phase. To render a good approximation of the signal, the stopping criterion is set to $T = 1000$ iterations. Let $x(t)$ be the signal and $g_{\gamma_i}(t)$ be the Gabor atom with $\gamma_i = \{s, u, \omega, \theta\}$ as parameters in Eq. (2.4). After T iterations, we have

$$x(x) = \sum_{i=1}^{T} \alpha_i \, g_{\gamma_i}(t) + R^T x. \tag{2.5}$$

The Time-Frequency Matrix (TFM) representation of $x(t)$ can be written as

$$V(t, f) = \sum_{i=1}^{T} \alpha_i \, WVG_{\gamma_i}(t), \tag{2.6}$$

where WVG_{γ_i} is the WignerVille distribution (WVD) of Gabor atom $g_{\gamma_i}(t)$. The WVD is a quadratic time-frequency representation in form of

$$W(t, f) = \frac{1}{2\pi} \int x(t - \tau/2) x^*(t + \tau/2) e^{-jf\tau} \, d\tau. \tag{2.7}$$

If signal $x(t)$ has more than one time-frequency component, its WVD will have cross-terms. However, given the decomposition of $x(t)$ in terms of Gabor atoms which consist of a single time-frequency component, $WVG_{\gamma_i}(t)$ in (2.6) will not have a cross-term interference. As a result, TFM $V(t, f)$, can be considered as an accurate representation of the spectrogram of the signal. Since only first T atoms are used, less significant time-frequency components are filtered out and the desired structural property of the energy distribution is captured in $V(t, f)$. Then, the Non-Negative Matrix Factorization

(NFM) is applied to $V(t,f)$ to obtain a more compact representation in terms of time and frequency:

$$V = WH, \tag{2.8}$$

where W and H capture the frequency and temporal structures of each component, respectively. One can reduce the redundant information in $V(t,f)$ by decomposing it into fewer components. Finally, the following four features are extracted.

1. *Joint TF moments.* The p^{th} temporal and q^{th} spectral moments are defined as

$$MO_{h_j}^{(p)} = \log_{10} \sum_n (n - \mu_{h_j})^p h_j(n), \tag{2.9}$$

$$MO_{w_j}^{(q)} = \log_{10} \sum_n (n - \mu_{w_j})^q w_j(n). \tag{2.10}$$

2. *Sparsity.* The measure of sparseness of temporal and spectral structures help in distinguishing between transient and continuous components. They are defined as

$$S_{h_j} = \log_{10} \frac{\sqrt{N} - \Big(\sum_n h_j(n)\Big) / \sqrt{\sum_n h_j^2(n)}}{\sqrt{N} - 1}, \tag{2.11}$$

$$S_{w_j} = \log_{10} \frac{\sqrt{N} - \Big(\sum_n w_j(n)\Big) / \sqrt{\sum_n w_j^2(n)}}{\sqrt{N} - 1}. \tag{2.12}$$

3. *Discontinuity.* The abrupt changes in the structure of temporal and spectral components are measured by the following parameters:

$$D_{h_j} = \log_{10} \sum_n h'^2_j(n), \tag{2.13}$$

$$D_{w_j} = \log_{10} \sum_n w'^2_j(n), \tag{2.14}$$

where $h'_j(n)$ and $w'_j(n)$ are the first order derivatives of temporal and spectral components, respectively.

4. *Coherency.* The coherency of the MP decomposition of a given signal, $x(t)$, can be evaluated as

$$CMP = \log_{10} \frac{\sum_{t=2}^{T} \alpha_t - \alpha_{t-1}}{E_x}, \tag{2.15}$$

where E_x is the total energy of signal $x(t)$.

Finally, LDA is used for classification.

There are justifications to the approach proposed in [17]. First, the WVD is a quadratic representation and so is energy (and in turn the spectrogram). By using the WVD of a single component, one obtains a cross-term free estimate of the spectrogram by retaining all useful properties of the WVD while leaving out its drawback. Second, the NMF yields a compact pair of vectors which contain important time-frequency components in the signal. Hence, features derived from these components tend to be characteristics of the underlying signal. When compared to MP-Gabor features, the first and second order moments estimated with this method are more reliable.

On the other hand, there are several weaknesses in this approach. First, there might be a problem with the discontinuity measure. The NMF results in non-unique decomposition. An intuitive initialization based on signal properties was adopted in [17]. However, it is not guaranteed that the discontinuity measure would be stable for signals of the same class as the order of spectral and temporal components in vectors W and H affect this measure. It would be better to sort the components before taking the first derivative of these quantities. Second, its computational complexity is way too high. One needs to perform the MP decomposition of a 3-second signal sampled at $F_s = 22.05$ kHz up to 1000 iterations. Moreover, all possible discrete points of scale, frequency, location and orientation parameters are needed. Given these conditions, each iteration would require

about $(6F_s + 1)M$ operations, where M is the number of atoms in the Gabor dictionary. The length of a 3-second signal $x(t)$ is $3F_s$, and an over-complete dictionary with at least $M = 4 \times 3F_s$ should be used. As a result, the total number of operations needed at each iteration would be about $72F_s^2 \approx 1.58$ million operations. It is desirable to implement the algorithm using the sub-framing processing scheme, yet this will result in a distorted estimate of long-term variations.

In [18], Ghoraani and Krishnan applied a nonlinear classifier called the Discriminant Cluster Selection (DSS) to the time-frequency features in [17]. The DSS uses both unsupervised and supervised clustering methods. First, all features, irrespective of their true classes, undergo an unsupervised clustering scheme. Resulting clusters are subsequently categorized as *discriminant* or *common* clusters. Discriminant clusters are dominated with majority membership from one single class while common clusters house features from all or multiple classes with no obvious champion-class. For a test signal, all features are first extracted from the signal. Then, each feature's membership is determined. Features belonging to common clusters are ignored. The final decision for a test signal is made based on the labels of discriminant clusters. Two schemes; namely, hard and soft/fuzzy clustering, are used in the last step. The crux of this algorithm is that it determines discriminant sub-spaces in the entire feature space. Each discriminant region is assigned to a single class. Given that a single test signal is represented by multiple features, its final labeling is done based on the cluster-membership relationship of its discriminant features.

The spectrogram offers a tool for visually analyzing the time-frequency distribution of an audio signal. This has inspired the development of visual features derived from the spectrogram of music signals [19, 62, 63]. The original application in [62] was texture classification, yet the plausible use for music instrument classification was mentioned. Souli and Lachiri subsequently used this method for ESR in [44]. They also proposed

another set of nonlinear features in [45]. In [45], non-linear visual features are extracted from the log-Gabor filtered spectrogram. The log-Gabor filtering is often used in image feature extraction. One polar representation of the log-Gabor function in the frequency domain is given by

$$G(r,\theta) = G_{radial}(r)G_{angular}(\theta), \tag{2.16}$$

where

$$G_{radial}(r) = e^{-\log(r/f_0)^2/2\sigma_r^2} \tag{2.17}$$

$$G_{angular}(\theta) = e^{-(\theta/\theta_0)^2/2\sigma_\theta^2} \tag{2.18}$$

are frequency responses for the radial and the angular components, respectively, f_0 is the center frequency of the filter, θ_0 is the orientation angle of the filter, and σ_r^2 and σ_θ^2 are the scale and the angular bandwidths, respectively. This method extracted features from the log-Gabor filtered spectrogram (instead of the raw spectrogram). Since no performance comparison was made between features obtained from the log-Gabor filtered spectrogram and the raw spectrogram in [44], the advantages and shortcomings of this approach need to be explored furthermore.

2.3 Limitations of Existing Methods

We conducted an in-depth survey on recent developments in the ESR field in this paper. Existing ESR methods can be categorized into two types: stationary and non-stationary techniques. The stationary ESR techniques are dominated by spectral features. While these features are easy to compute, there are limitations in the modeling of non-stationary sounds. The non-stationary ESR techniques obtain features derived from the wavelet transform, the sparse representation and the spectrogram. A set of features

with simplicity of stationary methods and accuracy of non-stationary methods is still a puzzle piece.

Chapter 3

Narrow-Band Time Frequency Representation

3.1 Introduction

Time-Frequency features seem to naturally arise from the fact that they decompose signal in both time and frequency space simultaneously. The concept of instantaneous frequency was first introduced by Gabor. Given this idea of instantaneous frequency, most information about a signal can be obtained using time-frequency decomposition.

3.1.1 Sparse Representation based Time-Frequency Features

In previous works [8, 43], time-frequency features have been successfully applied for ESR. The Basis Matching Pursuit (BMP) is a greedy algorithm used to obtain a sparse representation of signals based on atoms in an over-complete dictionary. Given signal x and an over-complete dictionary $D = [d_1, d_2, ...]$, BMP obtains the sparse representation of x on D as follows.

1. Initialize the residue at the 0^{th} iteration as $R^0 x = x$
2. For $t = 1$ to T
   (a) Select the atom with the largest inner product with the residue via

$$d_t = \max_i < R^{t-1}x, d_i > .$$

(b) Update the residue via

$$R^t x = R^{t-1} x - \alpha_t d_t,$$

where $\alpha_t =< R^{t-1}x, d_t >$ is the projection coefficient of $R^{t-1}x$ on d_t.

3. The BMP projection of x on D is given by

$$\hat{x} = \sum_{i=1}^{T} \alpha_i d_i$$

One stopping criterion for this algorithm is a fixed number of iterations (atoms), T. Another one is to use the energy of the residual signal, i.e. decomposition stops at t when $||R^{t-1}x||^2 <$ Threshold.

An over-complete Gabor dictionary consisting of frequency modulated Gaussian functions (called Gabor atoms) was used in [8]:

$$g_{s,u,\omega,\theta} = \frac{K_{s,u,\omega,\theta}}{\sqrt{s}} e^{-\pi(n-u)^2/s^2} \cos\left[2\pi\omega(n-u)+\theta\right], \tag{3.1}$$

where s, u, ω and θ are atom's scale, location, frequency and phase, respectively, and $K_{s,u,\omega,\theta}$ is a normalization constant so that $||g_{s,u,\omega,\theta}||^2 = 1$.

3.1.2 Limitations of previous approaches

In [8, 43], it was shown that 5 atoms are sufficient for good classification accuracy. However, as we will show in experiments section, this statement does not hold true if the database is much larger than that used in [8]. In particular, a lot of similar sounding sounds belonging to different classes, like plastic impact and wood impact sound, would have similar response for quite a few frequencies. In Figure 3.1 we show one sample

each for wood collision and plastic collision. Figure 3.2 shows top 5 atoms using MP over Gabor dictionary. Though the scale is different, frequency parameters are quite similar for the two cases. This can be confirmed from the Table 3.1 which shows mean scale and frequency parameters for both cases. It should be noted that scale parameters are few and are coarsely sampled, which in turn degrades the discriminatory power of these features. Similar problem arises with the features proposed in [43].

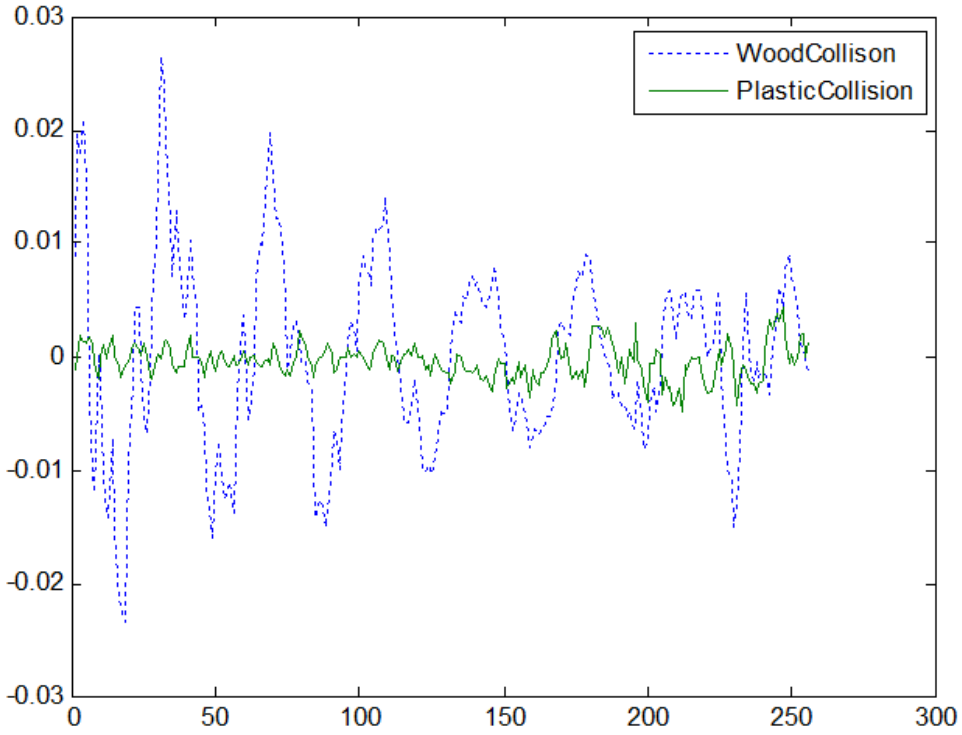

Figure 3.1: An example each from WoodCollision and PlasticCollision classes

One solution to overcome this problem would be to increase the number of atoms used. However, the final feature used for a frame of data sample is the mean of features over all the sub-frames. One can intuitively conclude tat this mean estimate would degrade in quality if there are too many atoms because such a recomposition would

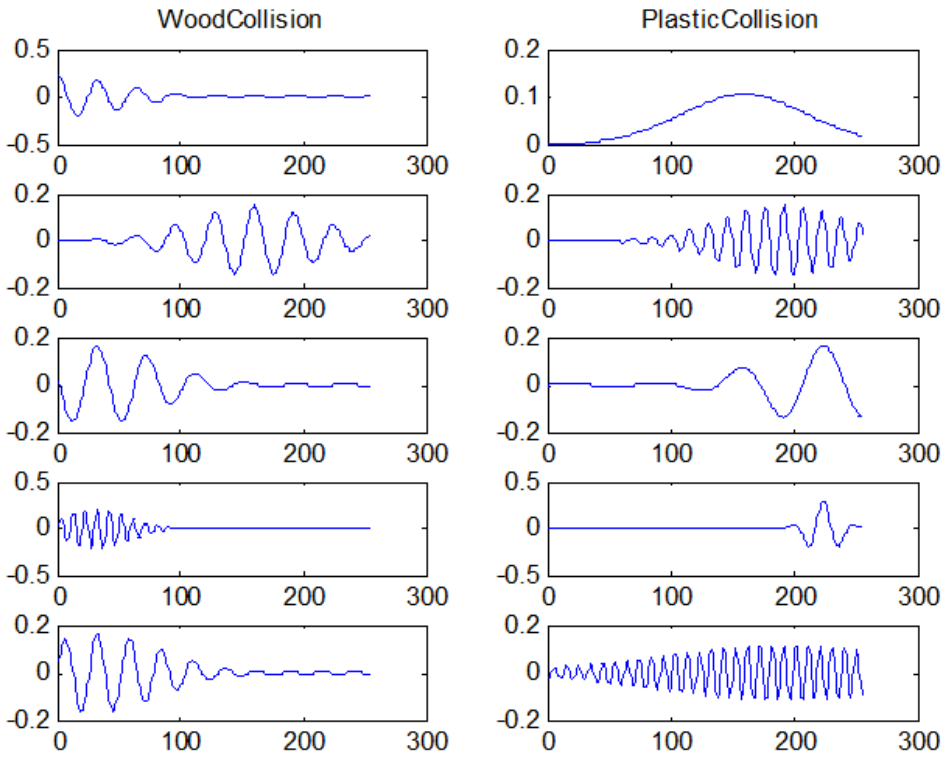

Figure 3.2: Time-Frequency decomposition of signals shown in Figure3.1

Table 3.1: Comparison of mean scale and frequency parameters for samples one each from WoodCollision and PlasticCollision

	Mean Scale	Mean Freq.
WoodCollision	115.2	0.0453
PlasticCollision	134.4	0.044

even represent finer details of the signal, which might vary a lot among samples of one class. Chu *et al.* also discuss this point, and state that the finer details can be "noisy" when homogeneity of a class is considered[8]. Hence, a better feature representation is needed.

3.2 Narrow-Band Time Frequency Representation

3.2.1 Motivation

Gabor representations are effective in capturing non-stationary characteristics of signals. Chu *et al.* showed this in their pioneering work[8]. In the previos section we pointed out some limitations of these features. MFCC also have been very effective for the task of ESR, as we will show in the Section 3.3. Hence, we believe it would be effective to combine the two features to obtain better results. This motivated us to propose new feature, Narrow-Band Time Frequency (NBTF)features.

3.2.2 Feature Extraction

Now we will introduce the NBTF features. First, signal x is filtered using a Mel-Filterbank with M bands. The motivation of using Mel-Filterbank comes from the fact that these filter banks are linearly spaced over Mel Scale, which in turn captures perceptual scale of pitches. Hence, we obtain N filtered signals:

$$x_m = x \ * \ MF_m \ \ \forall m \in \{1, \ldots, M\} \tag{3.2}$$

Here, MF_m is an Finite Impulse Response filter corresponding to m^{th} band on Mel-Scale. Then, for each filtered signal x_m, Matching Pursuit is used to obtain its sparse representation over Gabor dictionary D, as discussed in Section 3.1.1. Here, just as in [8], only first N atoms are extracted and mean and standard deviation of frequency and scale parameters for these 5 atoms are calculated. Let (μ_{sm},σ_{sm}) and ($\mu_{\omega i}$,$\sigma_{\omega i}$) denote

the mean ans standard deviation of scale and frequency parameters, respectively. Then, NBTF features for x can be given by:

$$NBTF(x) = \begin{bmatrix} TF(x_1) \\ \vdots \\ TF(x_M) \end{bmatrix} \quad \text{where } TF(x_m) = \begin{bmatrix} \mu_{sm} \\ \sigma_{sm} \\ \mu_{\omega m} \\ \sigma_{\omega m} \end{bmatrix} \tag{3.3}$$

Figure 3.3 shows the feature extraction process.

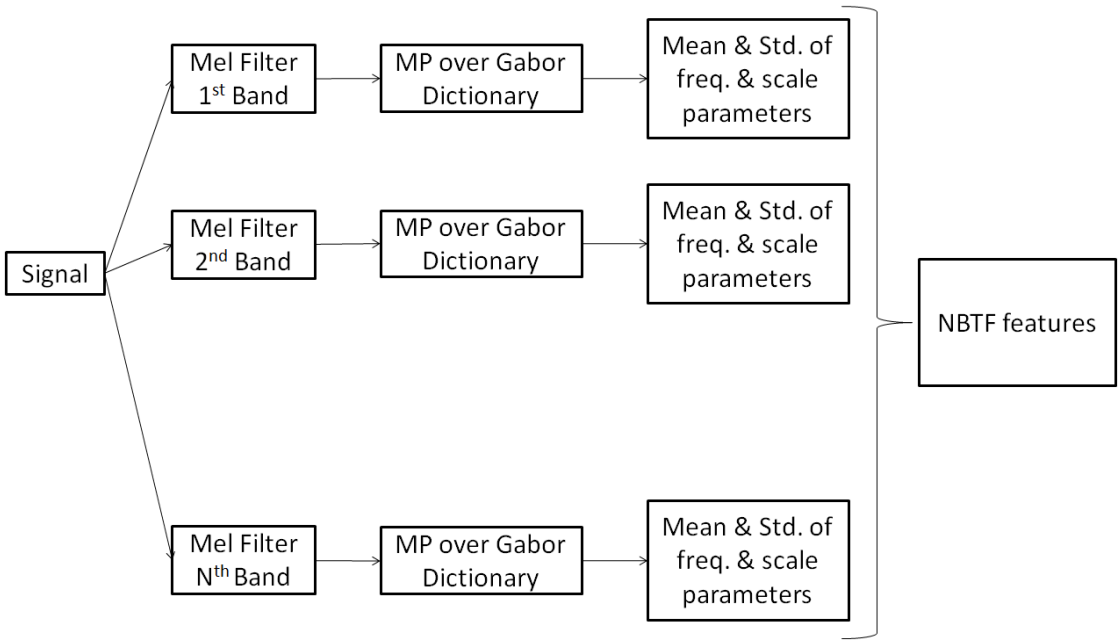

Figure 3.3: NBTF Feature Extraction

3.3 Experiments

3.3.1 Database and Experimental Setup

One major problem in the ESR field is the lack of a universal database. There are some consolidated acoustic databases for specific applications such as study of elephant calls[1] and acoustic for emotion stimuli[2] database. However, these databases consists of sounds either limited to an application or not directly related to environmental sounds. Papers in this field generally present their results with their own dataset consisting of an arbitrary number of environmental sound classes collected from various sources, mostly from the Internet. In the absence of a standard database, it is difficult to conduct a quantitative comparison of various approaches. Baseline classifiers with Mel-Frequency Cepstral Coefficients (MFCC) are often used to benchmark the performance of a new algorithm. However, due to significant differences in datasets of any two papers, such a performance benchmarking is futile. Hence, we built our own Environmental Sound Recognition Database (ESRD) from various sources. Most of our audio clips came from sound-effects library provided by Audio Network[3]. We also used the BBC Sound Effects Library[4], the Real World Computing Partnership's (RWCP) non-speech database [34], and freely available audio clips from various sources on the Internet[5][6][7]. Our database consists of 37 classes which are a mixture of sound events and

[1]Elephant Call Types Database http://www.elephantvoices.org/multimedia-resources/elephant-call-types-database.html

[2]International Affective Digital Sounds http://csea.phhp.ufl.edu/media.html

[3]http://www.audionetwork.com/sound-effects

[4]http://www.sound-ideas.com/bbc.html

[5]Find Sounds: Search the Web for Sounds http://www.findsounds.com/

[6]The Free Sound Project https://freesound.org/

[7]Royalty Free Sounds from Youtube http://www.youtube.com/

ambiance sounds. Table 3.2 shows these classes and the corresponding number of data points sampled from all the ESR audio clips. Here, we are showing the number of sampled data points instead of the total duration of clips in the database because this gives a more insightful view of the database. All audio clips were first down-sampled to 16KHz, 16-bit mono audio clips. These clips were then sampled to give data points for training and testing. Ideally, we wanted to have a single train/test data point to be of 6 sec. However, some events are short-lived, like those from the RWCP database (Metal Collision, Wood Collision, etc.). Hence, variable length sampling was used and it resulted in data points of 0.5-6 sec in length. Naturally, the number of sampled data points in ESRD gives a more comprehensive view about the database in such a scenario.

Table 3.2: Environmental Sound Recognition Database (ESRD)

(C1)AirplaneFlyBy	660	(C14)DogsBarking	577	(C27)Rubbing	500
(C2)AirplaneInterior	662	(C15)Fans/Vents	585	(C28)Snoring	459
(C3)ApplauseCheer	424	(C16)FireCrackle	697	(C29)Streams	1194
(C4)BabyCryFuss	842	(C17)Footsteps	786	(C30)Thunder	412
(C5)Bees/Insects	514	(C18)GasJetting	269	(C31)TrainInterior	980
(C6)Bells	456	(C19)GlassBreakCrash	715	(C32)Vacuum	524
(C7)Birds	1189	(C20)HelicopterFlyBy	916	(C33)Waterfall	792
(C8)BoosOhsAngry	621	(C21)MachineGuns	526	(C34)WhalesDolphins	510
(C9)CatsMeowing	392	(C22)Metal Collision	1000	(C35)Whistle	300
(C10)CeramicCollision	800	(C23)Ocean	322	(C36)Winds	956
(C11)Clapping	829	(C24)PaperTearCrumble	351	(C37)WoodCollision	1187
(C12)Coins	616	(C25)Plastic Collision	550		
(C13)Crickets	550	(C26)Rain	694		

We see from Table 3.2 that the database is non-uniform. In order to bring a sense of uniformity among various classes, we randomly selected a *maximum of* 400 *samples* to represent each class. 70% of the these samples were used for training while the remaining 30% were used for testing. To obtain reliable results, we repeated the experiments

30 times by randomly reselecting up to 400 data points for each class, and again randomly generating training and testing sets for these 400 data points. Note that some classes such as CatsMeowing(C9) and GasJetting(C18) have less than 400 samples to begin with and hence all data points are used for all trials without any sampling. We did not want to create bias by generating 400 samples by sampling with replacement. For each set of experiments, we used 5-fold cross validation to select classifier parameters. We used MATLAB's in-built routines for GMM and FFNN, and LIBSVM[6] for SVM.

3.3.2 Baseline Experiments

Performance comparison was made between ten selected methods listed in Table 3.3. Methods M1 uses sub-framing based processing scheme (see Section 2.1) wherein averaged MFCC features from all the sub-frames are used to represent a single data point. GMM and SVM were both used for multi-class classification, and with initial trials we notice that SVM performance better than GMM, so SVM was eventually used for comparison purposes. Methods M2-M9, and M11 are selected from various publications discussed before in Section 2.2.1 and Section 2.2.2. M10 is the proposed method, NBTF.

For all methods, we tried our best in strictly following the experimental setup as stated in original papers. However, we did have to make changes to the framework of certain methods as our database consists of variable length data-points. For example, in Method M8, authors use a sub-framing based processing scheme wherein the data from all sub-frames are concatenated to form the feature vector. Basically, this scheme assumes that the number of sub-frames is the same for all data points. For our variable length database, this is however not true. In order to comply with this requirement, we chose to replicate the data-point to form a 6 sec length data-point and used an appropriate tap-sized moving average filter to smooth the overlapping regions of replicates. Similar

Table 3.3: Selected Methods for comparison

Label	Method	Feature	Classifier	Dimensionality Reduction/ Feature Selection	Stationary(S)/ Non-Stationary(NS)
M1	N/A	MFCC	SVM	No	S
M2	Karbasi *et al.* [23]	SDF	K-NN	DCT	S
M3	Valero and Alias [50]	NB-ACF	SVM	No	S
M4	Valero and Alias [51]	Gammatone Wavelet	SVM	No	NS
M5	Umapathy *et al.* [48]	WPT	SVM	LDA	NS
M6	Chu *et al.* [8]	MP-Gabor	SVM	No	NS
M7	Sivasankaran and Prabhu [43]	Modified MP-Gabor	SVM	No	NS
M8	Khunarsal *et al.* [24]	Spectrogram	FFNN	No	NS
M9	Souli and Lachiri [45]	Log-Gabor filtered Spectrogram	SVM	Mutual Information	NS
M10	Proposed Feature	NBTF	SVM	No	NS
M11	Wang *et al.* [55]	Non-Uniform Freq. Map (NUMAP)	SVM	PCA+LDA	NS

adjustments were made in other experimental set-ups to fit our database. We show the pertinent details of these methods in Table 3.3.

3.3.3 Results and Discussion

In order to obtain stable results, we did 30 trials of training and testing as described in Section 3.3.1. Classification accuracy for class CN ($N = 1, 2, \ldots, 37$) is defined as the percentage of test data samples of class CN correctly classified. Average of the classification accuracies of all classes is used as a metric for single trial. This is done to avoid over-shadowing classification accuracies of classes with less than 400 samples. Finally, classification accuracies of all trials is averaged to quantify the performance of a single method. Figure 3.4 shows this averaged overall classification accuracy over 30 trials. For NBTF, we used $M = 5$ Mel-Scale Filterbanks.

The proposed non-stationary method M10 gives the best performance with average classification accuracy of 79.84%. The second best classification accuracy of 76.74% is achieved M1, a stationary method. Another recent non-stationary method M11 gives

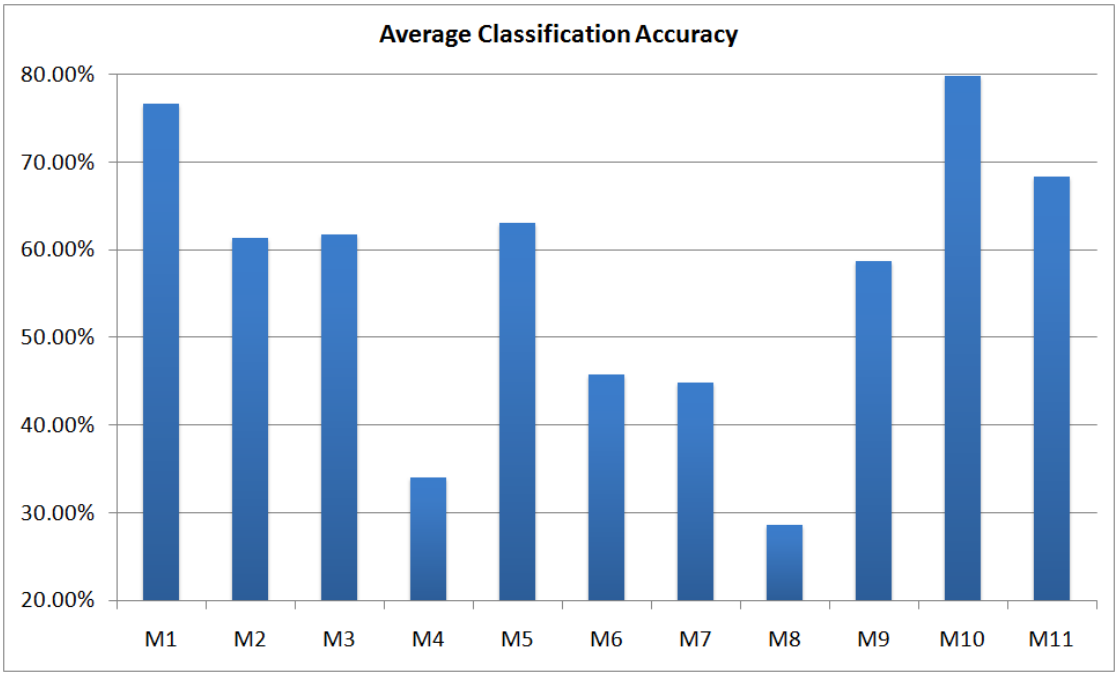

Figure 3.4: The averaged classification accuracies over 30 trials.

good performance, with average classification accuracy of 68.34%. Remaining two stationary methods - M2 and M3, and two non-stationary methods - M5 and M6, all give comparable performances. Surprisingly, M4, M6, M7, M8 despite being complex features, give poor performance. Figure 3.5 shows the average classification accuracy over all trials. It is clear that even when single instantiations are considered, M10 performs best, followed by M1 and then M11. We performed two popular tests to verify the statistical significance of classification results. Consider to methods A and B, such that n_{ab} denotes number of samples misclassified by both A and B, $n_{ab'}$ denotes number of samples misclassified by only A, $n_{a'b}$ denotes number of samples misclassified by only B, and $n_{a'b'}$ denotes number of samples correctly classified by both A and B. Let $n = n_{ab} + n_{ab'} + n_{a'b} + na'b'$ be the total number of samples in the test set. Also consider the null hypothesis that the two methods have same error rate. Given, this setting,

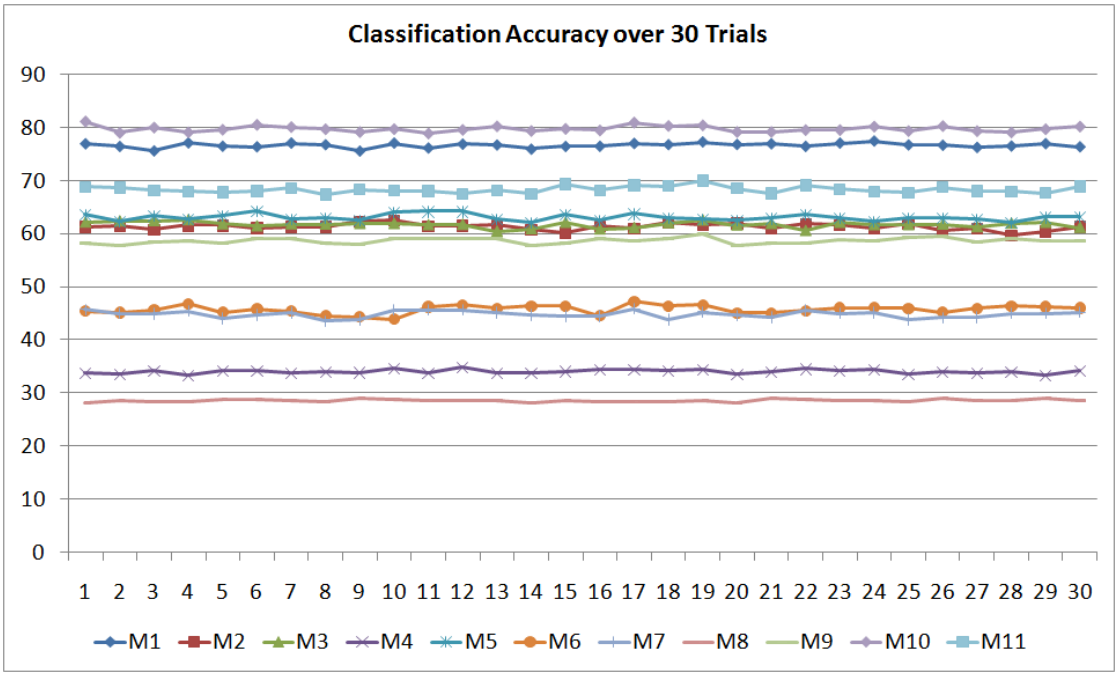

Figure 3.5: Classification Accuracies for 30 trials.

McNemar's test statistic M_{stat} shown below, approximately follows a χ^2-distribution with 1 degree of freedom:

$$M_{stat} = \frac{(|n_{ab'} - n_{a'b}| - 1)^2}{n_{ab'} + n_{a'b}} \tag{3.4}$$

Thus, for a p-value of 0.0001, null hypothesis is rejected if M_{stat} is greater than $\chi^2_{1,0.9999} = 15.1367$. It should be noted that McNemar's test can only be performed for one trial. Thus, with 30 trials, we get 30 different M_{stat} for each pair (A,B). Table 3.4 shows McNemar's test statistic for all pairs, and Table 3.5 shows pairs which frequently fail the test over 30 trials.

The other test we performed was the paired t-test. With this test, we aim to evaluate statistical significance of classification results over all the 30 trials[14]. In fact, the choice of 30 trials was motivated by the fact that at least 20 trials are necessary for this

Table 3.4: McNemar's Test Statistic for 1 of 30 trials

	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
M1	233.72	278.81	1639.31	261.27	1049.83	1001.72	1008.63	395.81	38.47154	96.02
M2		0.16	756.55	2.18	255.43	249.85	265.24	9.30	412.2574	55.06
M3			1033.35	1.61	315.25	318.53	566.35	14.18	477.1636	59.69
M4				978.10	167.05	179.60	637.70	637.73	1858.082	1303.60
M5					353.93	348.28	488.35	22.82	428.8729	42.55
M6						0.01	304.82	194.68	1305.812	586.54
M7							187.56	182.87	1289.186	585.26
M8								328.80	589.03	603.22
M9									594.652	112.86
M10										236.65

Table 3.5: Class-pairs that Frequently Failed McNemar's Test over 30 Trials

Class Pair	**No. times McNemar's test fails**
M2 - M3	30
M2 - M5	30
M2 - M9	23
M3 - M5	30
M3 - M9	19
M5 - M9	8
M6 - M7	30

test. In the same setting as before, let $e^i_A = (n^i_{ab'} + n^i_{a'b})/n^i$ and $e^i_B = (n^i_{ab'} + n^i_{a'b})/n^i$ be the error rates for i^{th} trial, then the difference between the two error rates over all trials follow Student's t-distribution with $n-1$ degrees of freedom. The T_{stat} for this test can be given by:

$$T_{stat} = \frac{\bar{e}\sqrt{n}}{\sqrt{\frac{\sum_{i=1}^{n}(e^i - \bar{e})^2}{n-1}}} \tag{3.5}$$

where $e^i = e^i_A - e^i_B$, and $\bar{e} = \frac{1}{n}\sum_{i=1}^{n} e^i$. Thus, with p-value of 0.0001 for a two tailed test, the null hypothesis can be rejected if absolute value of T_{stat} is greater than $t_{29,0.99995} = 4.5305$. Table 3.6 sows the t-statistic for each pair of classes for 30 trails.

Table 3.6: Paired T-Test Statistic for 30 trials

	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
M1	119.32	126.97	425.73	98.66	208.54	259.97	487.09	166.19	-91.90	63.76
M2		-3.08	229.04	-12.41	75.25	102.69	270.36	17.62	-119.67	-46.85
M3			210.86	-8.60	93.80	106.58	291.72	21.56	-120.14	-45.35
M4				-283.66	-75.49	-88.91	70.51	-234.10	-439.12	-290.40
M5					100.22	145.40	309.05	31.81	-132.78	-33.85
M6						6.30	108.25	-85.17	-218.40	-137.36
M7							122.57	-95.63	-256.43	-164.58
M8								-280.48	-429.75	-311.93
M9									-174.31	-69.37
M10										100.69

Figure 3.6 shows the performance comparison between the three methods - M6, M7 and M10, all based on the MP-Gabor features. Performance improvement by the proposed M10 scheme is clear in this figure. M10 gives higher classification accuracy for all classes, except C20 as shown in Table 3.2. The modified MP-Gabor feature [43] does not perform better than the MP-Gabor feature in [8] in most classes, except for classes like bells(C6), footsteps(C17), where sounds are clearly band limited. This is attributed to that the modified MP-Gabor feature [43] allows higher resolution in high-energy bands. From both Tables 3.4 and 3.6 it can be confirmed that the improvement of M10 over M6 and M7 is statistically significant.

Figure 3.7 facilitates the comparison between the 4 methods - M2, M3, M5 and M9 all of which have similar performance. Non-stationary methods M5 gives a slightly better performance than the others. The two non-stationary methods, M5 and M9 are strikingly balanced with respect to their favored class-wise performances. All the classes can be divided into two distinct balanced groups where one outperforms the other by a considerable margin. However, it should be pointed out that there seem to be some characteristically definable common denominator to these two groups. For example, if we consider impact sounds from RWCP database, M5 performs better for ceramic

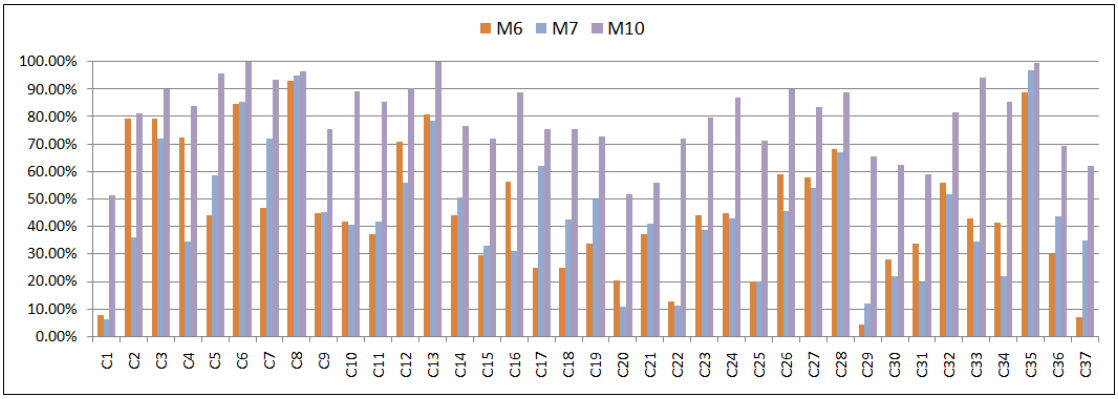

Figure 3.6: Comparison of averaged classification accuracies of M6, M7 and M10.

collision(C10) and wood collision(C37), where as M9 performs better for metal collision(22) and plastic collision (C25). C22 and C25 sounds are sharper than those of C10 and C37. It seems only natural that log filtered spectrograms would have reasonable resolution even at high frequencies. When M2 and M3 are compared, it can be seen that M2 performs better than M3 for more than 70% of the classes, still overall accuracy of both is comparable. This is because M2 performs really poorly for most of RWCP classes which are short-burst sounds. For these same classes, M3 gives a very good performance. Hence, we can again consider these two features as complementary to each other. Despite all these subtle differences, the results of all these four classes are not statistically different. In other words, the null hypothesis that these methods have similar average performance cannot be rejected, specifically using McNemar's test as shown in Table 3.4 and 3.5. It is interesting to see that the test fails very often for 30 trials. T-test, on the other hand, suggests that only M2 and M3 have statistically similar results. However, it should be noted that assumptions underlying T-test do not always hold true, and it is very susceptible to Type I error[14].

Finally, we would also like to compare the performance of top three methods - M1, M10 and M11. Class-wise performance for these three methods is shown in Figure 3.8.

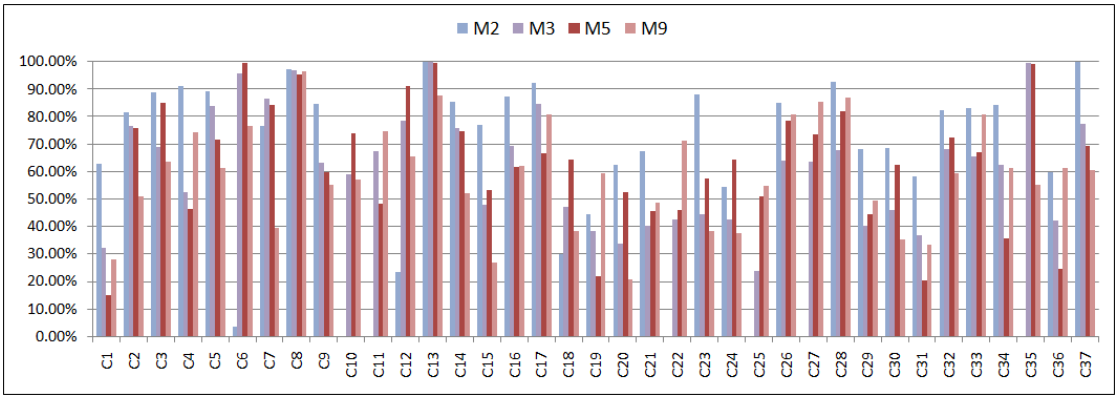

Figure 3.7: Comparison of averaged classification accuracies of M2, M3, M5 and M9.

The proposed method, M10, consistently performs better than M11, for both stationary and non-stationary sounds, except for few classes like footsteps(C17), thunder(C30) and vacuum(C32). In fact, for these three classes, M11 performs better than M1 as well. M1 closely follows performance of M10. In general, M1 performs better for classes that came from RWCP database, i.e. impact sounds. It also performs better for wideband classes such as airplaneInterior(C2), trainInterior(C31) and vacuum(C32). This leads us to conclude that simple yet powerful feature, MFCC, despite being a stationary feature, can handle even non-stationary classes very efficiently. The overall classification accuracy is still in late $70's$ and hence, this leaves room for development of better features.

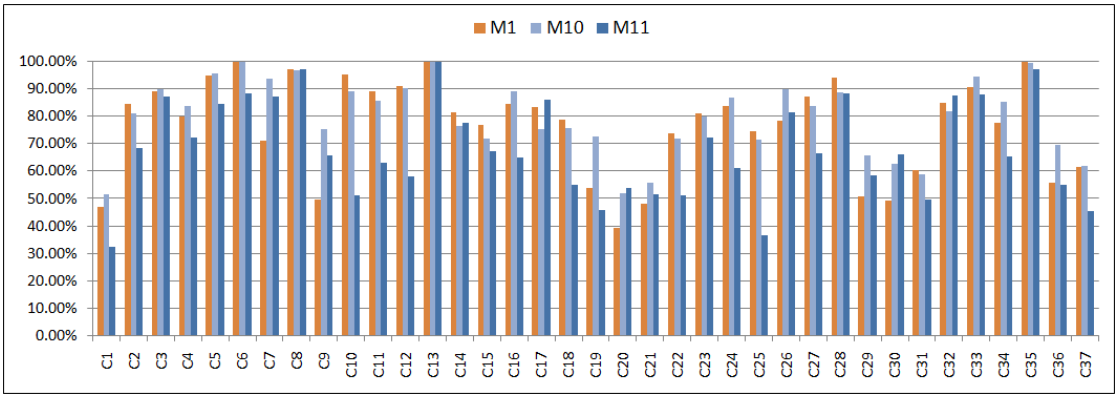

Figure 3.8: Comparison of averaged classification accuracies of M1, M10 and M11.

It is worthwhile to point out that both spectrogram-based methods perform poorly. M8 uses the spectrogram as features directly whereas M9 uses the spectrogram after filtering it through the Log-Gabor filter bank. Both methods seem to be plagued by the curse of dimensionality. However, performance of M9 significantly improves after dimensionality reduction. The same cannot be applied to M8 as this beats the authors' original motivation of directly using entire spectrogram.

We studied the effect of changing number of Mel-Filterbanks on the performance of NBTF features. We also studied the effect of using partial NBTF features, like using only mean scale values, using only mean frequency values, etc. Figure 3.9 shows the performance for various Cases (See Table 3.7 for details) and different number of bands in Mel-filterbanks. As we increase the number of bands, the performance improves for

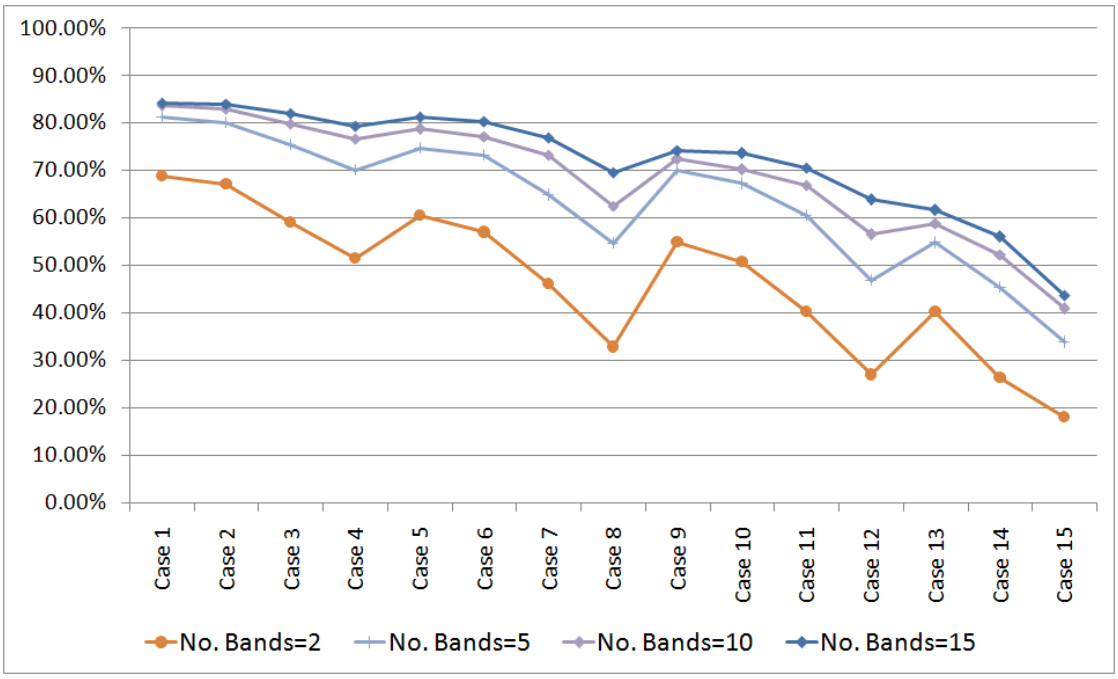

Figure 3.9: Effect of varying number of Mel-Filterbanks on Performance of NBTF (M10). See Table 3.7 for details on Cases

each Case. However, we feel that the improvement after 5 bands is not worth the trouble

Table 3.7: Cases with Partial NBTF features

	Mean Freq. ($\boldsymbol{\mu_\omega}$)	Std. Freq. ($\boldsymbol{\sigma_\omega}$)	Mean Scale ($\boldsymbol{\mu_s}$)	Std. Scale ($\boldsymbol{\sigma_s}$)
Case 1	✓	✓	✓	✓
Case 2	✓	✓	✓	
Case 3	✓	✓		✓
Case 4	✓	✓		
Case 5	✓		✓	✓
Case 6	✓		✓	
Case 7	✓			✓
Case 8	✓			
Case 9		✓	✓	✓
Case 10		✓	✓	
Case 11		✓		✓
Case 12		✓		
Case 13			✓	✓
Case 14			✓	
Case 15				✓

because the computational complexity increases with increase in number of bands and also the feature dimension increases. Hence, we had used $M = 5$ bands for all of the previous discussion. It can be seen that for almost all cases where mean frequency values are used, performance is higher than those where they are not used, with the exception of Case 8 and Case 7. This shows that mean frequency values are very important for classification, however, they by themselves are not sufficient. Also, standard deviation of scale values are least important of all, and removing them reduces the accuracy by not more than 1% for any number of bands. In fact, the curves show clear consistency in relative performance for different cases over any number of bands.

Finally, we used MFCC and NBTF features together. Figure 3.10 shows the performance of these features together with varying number of bands. The overall classification accuracy reaches in the neighborhood of 85%, which is much more than those of individual features.

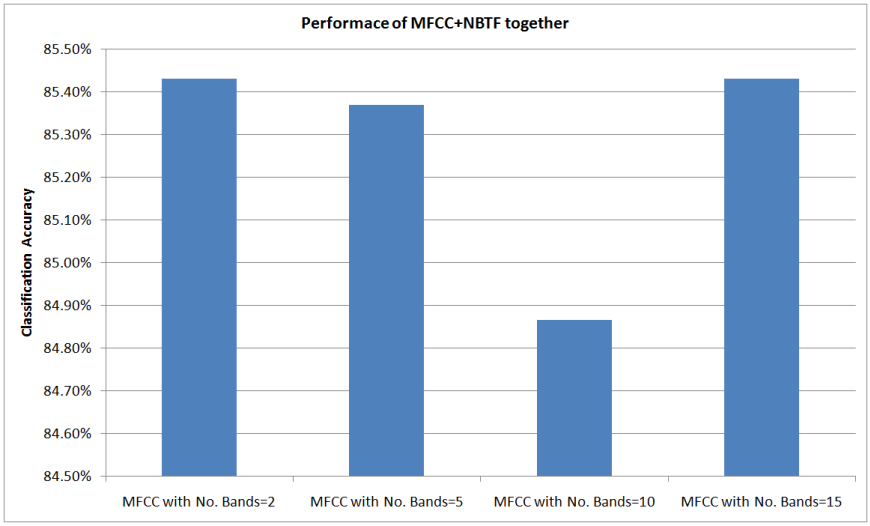

Figure 3.10: Performance of MFCC and NBTF when used together, with varying number of Mel-Scale bands

3.3.4 Conclusion

In this work, we compared the performance of several state of the art approaches on a common platform using our own ESR database. We provide a detailed critique on the performance of these features. We also proposed a new set of features, Narrow Band Time Frequency features and show it's superior performance as compared to all benchmark approaches. We were able to achieve an improvement of 4% as compared to the best feature in benchmark models, i.e. MFCC. Finally, an accuracy of about 85% is achieve when MFCC and NBTF ar used together, implying that the two features are complementary to each other.

Chapter 4

Environmental Sound Recognition using Multi-Classifier System

4.1 Introduction

In machine learning and pattern recognition, there are simple set of unwritten rules one follows to solve a classification problem. Raw data is often not directly usable for supervised learning problems; for example a, high dimensional data is hard to model. Hence, it is customary to extract meaningful features from raw data, which exhibit unique characteristics that better represent homogeneity and inhomogeneity amongst intra and inter group elements of dataset, respectively. If the dimensionality of these features is high, feature selection or reduction methods are used to control complexity of learning algorithms. Next, different classifiers are learned to identify various groups amongst the dataset. Finally, the most robust of these models is selected for the pattern recognition task at hand. Robustness can be ensured by various methods like holding test data, cross-validation, etc. However, such measures of robustness cannot guarantee that optimal model has been selected for all unseen data. One often wonders if there is a way to select the best model. Occams Razor Principle of Parsimony tries to answer this complicated question: given two classifiers that perform equally well on the training set, it is asserted that the simpler classifier may do better on test set. Defining simplicity is not a trivial task either. Each learning model comes with a set of assumptions and thus, bias as well. Given a finite set of data, each model converges to a different solution and fails

under different circumstances. Validation error is itself a random variable and cannot be assessed in true sense. Hence, making an optimal selection from choice of multiple available models is not an easy task.

In our day to day lives, when at crossroads, we often take opinions from several people when making important decisions. The motivation comes from the fact that different people give opinions from their perspective and we like to asses a problem from all angles before we make a decision. Similarly, given a dataset, different learning models will have different opinions based on certain underlying assumptions. Instead of making a decision based on single model, why not combine the opinions of various models to give a more robust solution? This school of thought has been embraced by scientific communities since late seventies. Over the years, several names have been given to this approach - composite classifier systems, mixture of experts, pandemonium system of reflective agents, consensus aggregation, classifier fusion, classifier ensembles, combination of multiple classifiers, ensemble learning, and many more. This ideology has also been used in practical systems such as Netflix challenge which first started in 2006.

Keeping this ideology in mind, in this work we build a learner based on multi classifier system (MCS) to significantly improve the performance of Environmental Sound Recognition (ESR). In the following sections, we first motivate MCS approach as against classical feature fusion approaches. Next we will provide a brief overview of MCS, and discuss role of diversity in it. Finally, we would present the proposed model, followed by rigorous experimental evaluation and discussion.

4.2 Feature Fusion Vs MCS

In pursuit for superior performance of ESR, several features have been proposed. However, as shown in previous chapter, none of them is single handedly capable of providing

robust results. More than often, when proposing a new method, authors would conclude that *combining* proposed features with some popular features would improve the classification accuracy of the system. This act of combination is nothing but feature fusion.

Feature fusion can also be viewed from different perspectives. For example, feature selection algorithms aim to find best set of features amongst various available options. Feature transformation methods, on the other hand, project features into more discriminatory sub-spaces. Nonetheless, feature transformation methods can also be considered as a way of feature fusion. Feature fusion has been extensively studied in the past, and yet there is no single global winner which is applicable to all learning problems.

In its simplest form, feature fusion simply refers to combining features from various methods to give a higher dimensional feature vector. However, this approach has many drawbacks. For example, combining several features together would greatly increase the dimensionality of feature space. Given a finite set of data points, now we would have a higher dimensional representation with sparsely sampled data points as compared to individual feature spaces. Also, different features are extracted under different framework-based rules which makes feature fusion impossible. For example, some features could use framing based processing, while others might use sub-framing based processing (see Section 2.1 for more details). Hence, due to the nature of processing parameters and methods, it may be impossible to combine different features. However, under the same circumstances, it is possible to formulate a MCS which would be able to use decisions from such fundamentally different processing schemes.

Following table 4.1, using one such MCS, shows the superior performance of MCS over feature fusion for ESR. The table shows the difference between Confusion Matrix for MCS and that of Feature Fusion approach. In this experiment, only six classes were used for comparison purposes. These classes were selected based on their generally poor performances in experiments in Chapter 3. These are the worst performing classes (see

Section 2.1 for more details). MCS used here will be discussed in detail in Section 4.4. It can be seen that the average improvement of MCS over Feature Fusion is about 16%. Except for Thunder, MCS improves classification accuracy for all the classes.

Table 4.1: Difference between Confusion Matrix for MCS and Feature Fusion approach. Positive number in a cell indicates higher MCS value as compared to corresponding Feature Fusion value. The average improvement of MCS over Feature Fusion is 16.25

AirplaneFlyBy	27.50	-4.17	1.67	-20.83	0.00	-4.17
Fans	6.67	5.00	2.50	-12.50	-2.50	0.83
HelicopterExterior	-1.67	-17.50	40.00	-4.17	-7.50	-9.17
Thunder	14.17	-5.00	0.00	-10.83	5.00	-3.33
TrainInterior	4.17	-6.67	0.00	-5.83	10.00	-1.67
Winds	3.33	-3.33	-15.00	-4.17	-6.67	25.83

4.3 Brief Overview of MCS

As mentioned before, MCS have been referred to by several names in the literature. Going by the distinction provided in Woods *et al.* [59], MCS can be broadly classified into two categories: classifier selection and classifier fusion. This categorization is based on how the classifiers are used to give a collective decision. There is another way to categorize MCS, that is based on whether data set is decomposed or features are decomposed to give statistically significant learning models: horizontal decomposition and vertical decomposition.

4.3.1 Classifier selection Vs Classifier Fusion

MCS can be broadly categorized into classifier selection models or classifier fusion models. Classifier selection models work on the principle that different classifiers have different accuracies in different regions of the feature space. Hence, given a data point,

the goal is to select a classifier which is most likely to give correct prediction. Thus, learning entails recognizing "expert" classifier in different regions of feature space. By contrast, in classifier fusion methods, opinions of all experts are used to make a prediction. Classifier fusion methods can again be divided into three categorizes based on how the fusion is carried out:

1. Based on Class Labels: In these methods, crisp class labels are directly combined to give final prediction. Methods such as voting methods, knowledge space methods fall under this category.

2. Based on Class Rankings: Instead of using crisp labels, these methods use class rankings. Borda count, logistic regression are some examples of such systems.

3. Based on soft\fuzzy outputs: Instead of using crisp labels, or discreet rankings, these methods use soft\fuzzy score outputs from classifiers which are then combined in fusion stage to get final prediction. The soft scores could also be probabilities or a measure of belief that a given data point belongs to a particular class with a certain degree of belief.

There are models which combine the two approaches like hierarchical mixture of experts. It should be noted that there is no known way to gauge which scheme will perform better without realizing both the schemes over test data. However, high "diversity" in data reasonably guarantee that classifier fusion will work.

4.3.2 Role of Diversity in MCS

MCS does not always outperform feature fusion models. It has been an open research question as to when it is beneficial to prefer one over the other. One of the commonly invoked reasoning is that if the different experts/classifiers provide "diversity" in error,

then MCS should be fruitful. However, concept of "diversity" greatly varies amongst scientific community. Brown et al. [5] provide an extensive overview of concept of diversity.

Diversity in classification problems is even more vague as compared to regression problems. For regression problems, the two most common perspectives are ambiguity decomposition and bias-variance decomposition, both of which are related. However, for classification, due to highly non-convex loss functions, it is hard to quantify diversity without actually performing exhaustive search of various hypothesis and comparing to quantitative propositions of diversity.

We will present some commonly used diversity measures. Consider two hypothesis, H_i and H_j, such

- a is the number of instances for which both H_i and H_j are correct.
- b is the number of instances for which H_i is correct and H_j is incorrect.
- c is the number of instances for which H_i is incorrect and H_j is correct.
- d is the number of instances for which both H_i and H_j are incorrect.

Then, following three are popular measure diversity:

1. Correlation Measure:

$$\rho_{i,j} = \frac{|ad - bc|}{\sqrt{(a+b)(c+d)(a+c)(b+d)}} \tag{4.1}$$

The correlation takes values between 0 and 1, with maximum diversity achieved for $\rho_{i,j} = 0$. There is an alternate definition of correlation measure and is be given by:

$$\rho_n = \frac{nN^f}{N - N^f - N^r - nN^f} \tag{4.2}$$

where, ρ_n is diversity of the entire MCS system (with two or more experts), N^f is the number of training data failed to be classified by all classifiers, N^r is the number of instances in training data correctly classified by all classifiers, N is the total number of training samples, and n is the number of classifiers. The smaller the correlation, the better the MCS performance. Table 4.2 shows the correlation diversity measure for various methods with reference to ESR database. It is clear there is sufficient diversity amongst various methods.

Table 4.2: Correlation Diversity Measure

	M1	M2	M3	M4	M5	M6	M7	M8	M9
M1	1.000	0.061	0.265	0.236	0.320	0.278	0.236	0.232	0.480
M2	0.061	1.000	0.132	0.174	0.100	0.099	0.088	0.042	0.057
M3	0.265	0.132	1.000	0.424	0.379	0.242	0.259	0.193	0.330
M4	0.236	0.174	0.424	1.000	0.331	0.266	0.309	0.212	0.251
M5	0.320	0.100	0.379	0.331	1.000	0.241	0.236	0.151	0.342
M6	0.278	0.099	0.242	0.266	0.241	1.000	0.379	0.226	0.339
M7	0.236	0.088	0.259	0.309	0.236	0.379	1.000	0.185	0.301
M8	0.232	0.042	0.193	0.212	0.151	0.226	0.185	1.000	0.209
M9	0.480	0.057	0.330	0.251	0.342	0.339	0.301	0.209	1.000

2. Q-Statistic:

$$Q = \frac{ad - bc}{(ad + bc)} \tag{4.3}$$

Q is positive is same instances are correctly classified by both classifiers, and negative values otherwise. Maximum diversity, hence, is achieved for $Q = 0$. Table 4.3 shows the Q-statistic diversity measure for various methods. Again, it is clear there is sufficient diversity amongst various methods.

Table 4.3: Q-Statistic Diversity Measure

	M1	M2	M3	M4	M5	M6	M7	M8	M9
M1	1.000	0.146	0.561	0.623	0.647	0.634	0.547	0.505	0.829
M2	0.146	1.000	0.271	0.379	0.210	0.203	0.181	0.087	0.126
M3	0.561	0.271	1.000	0.839	0.678	0.479	0.509	0.384	0.628
M4	0.623	0.379	0.839	1.000	0.696	0.521	0.589	0.448	0.588
M5	0.647	0.210	0.678	0.696	1.000	0.480	0.470	0.308	0.645
M6	0.634	0.203	0.479	0.521	0.480	1.000	0.664	0.441	0.683
M7	0.547	0.181	0.509	0.589	0.470	0.664	1.000	0.366	0.618
M8	0.505	0.087	0.384	0.448	0.308	0.441	0.366	1.000	0.431
M9	0.829	0.126	0.628	0.588	0.645	0.683	0.618	0.431	1.000

3. Disagreement $D_{i,j}$ and double fault measure $DF_{i,j}$:

$$D_{i,j} = b + c \tag{4.4}$$

$$DF_{i,j} = d \tag{4.5}$$

High values for both these quantities indicate high diversity. High diversity in ESR database can be seen from Table 4.4 where the normalized disagreement measure is shown.

4.3.3 Horizontal Vs Vertical Decomposition

Essential goal of any MCS system is to improve accuracy of a single classifier system. In order to do so, individual weak learners can work on either subset of feature dimensions or subset of feature space. These methods are very different from each other.

When feature space is divided into subspaces, each weak learner is trained to guarantee improved accuracy in its own subspace. This type of decomposition is known as

Table 4.4: Disagreement Diversity Measure

	M1	M2	M3	M4	M5	M6	M7	M8	M9
M1	0.00	41.10	32.56	49.41	29.83	40.80	42.51	35.67	20.73
M2	41.10	0.00	40.87	45.95	42.11	46.34	46.85	45.92	42.72
M3	32.56	40.87	0.00	34.63	29.00	39.46	38.63	38.63	30.43
M4	49.41	45.95	34.63	0.00	39.34	35.95	33.96	43.02	45.67
M5	29.83	42.11	29.00	39.34	0.00	39.69	39.92	40.48	29.55
M6	40.80	46.34	39.46	35.95	39.69	0.00	30.80	39.76	36.37
M7	42.51	46.85	38.63	33.96	39.92	30.80	0.00	41.75	38.07
M8	35.67	45.92	38.63	43.02	40.48	39.76	41.75	0.00	37.06
M9	20.73	42.72	30.43	45.67	29.55	36.37	38.07	37.06	0.00

Horizontal Decomposition. Here the subspaces could be mutually exclusive or overlapping. If regions are simply divided by dividing feature space, then classifier selection would be a good strategy. Instead of simply dividing the data points, Horizontal Decomposition can also be achieved by oversampling data points from a particular region. This is the case in popular methods like AdaBoost.

On the other hand, we can also divide feature dimensions into multiple subsets, and train a weak learner for each subset. Thus, each weak learner uses all the data points, but only certain projections of those data points on smaller subspaces. The motivation for this is that different projections of the data would provide different degree of separability among various classes, thereby creating diversity in decisions. Such a decomposition is known as horizontal decomposition. Random subspace method and random forest are two examples of horizontal decomposition methods.

4.4 MCS for ESR

In Section 4.3, we briefly introduced various MCS systems. It is evident that there are many ways to build a MCS. First, we need to select one of the two approaches - Classifier Fusion and Classifier Selection. Classifier fusion requires competitive experts where as classifier selection requires complementary experts. In Section 4.3.2, we showed that there is sufficient diversity in our base classifiers. Also, the correlation coefficients are not small (table). Thus, it is safe to conclude that given the choice of vertical decomposition of feature space, classifier fusion is the way to proceed. However, for a fair comparison, we will also include one algorithm for classifier selection method. We proposed several MCS systems which we will discuss in the following subsection, along with MCS systems which we will use for benchmarking our proposed models. Following notations would be used to assist the discussion:

- M classes with labels - $\{1, \ldots, M\}$
- A data sample t with label $L(t) \in \{1, \ldots, M\}$
- N feature sets - $\{f_1, \ldots, f_N\}$. Each feature set is supposed to capture some aspect of data, and is of finite dimensions. Formally:

$$f_i \in R^{d_i}, d_i < \infty, i \in \{1, \ldots, N\} \tag{4.6}$$

- N classifiers corresponding to each feature set - $\{\Psi_1, \ldots, \Psi_N\}$, where

$$\Psi_i : f_i \rightarrow \mathbb{M}, \mathbb{M} \equiv \{1, \ldots, M\} \tag{4.7}$$

4.4.1 Benchmark Models

Dynamic Classifier Selection with Local Accuracy (DCS-LA)

DCS-LA, first was proposed by Woods et al. [59], is a classifier selection MCS system which we would use for benchmarking. As the name suggests, the main idea is to define local accuracy for each classifier in the entire feature-space. As in [59], we use K-NN to define local accuracy. Given a test sample, local accuracy of a classifier is defined as the percentage of K-nearest training samples that are correctly classifier. Finally, the classifier with highest accuracy in that region is selected. It should be noted that in this work, we use different features for base classifiers as against using different classifiers with same features.

DCS-LA can be describe as follows:

1. Given a test sample t and corresponding feature set f_i, find K-nearest neighbors, $KNN(f_i(t)) = \{f_i(t_1), \ldots, f_i(t_K)\}$, from training data.

2. Calculate local accuracy for classifier corresponding to each feature set:

$$Acc(f_i(t), \Psi_i) = \sum_{j \in KNN(f_i(t))} \delta(L(t_j), \Psi_i(t_j)) \tag{4.8}$$

   where

$$\delta(i, j) = \begin{cases} 1, & \text{if} i = j \\ 0, & \text{otherwise} \end{cases} \tag{4.9}$$

3. Assign label of classifier with maximum accuracy in local region:

$$\Psi_{DCS-LA}(t) = \arg_i \max Acc(f_i(t), \Psi_i) \tag{4.10}$$

Behavior Knowledge Space (BKS)

BKS approach relies on the crisp labels. Given M classes and N classifiers, we can have a possible of M^N combinations of outputs from training data. Hence, one can explicitly identify which combination corresponds to which true label in the training data. A BKS unit corresponds to a point in M^N space and has following three types of data

- Total number of samples in training data which share that BKS unit
- Number of samples for each of the M classes out of total number of samples
- The best representative class, i.e. the class with maximum number of samples falling in that BKS unit

Given a test sample, we assign it the representative class of the BKS unit the test sample corresponds to. It should be noted that in original algorithm, one uses an additional decision, rejection, i.e. the unseen sample does not belong to any of the classes. However, in our work, we do not consider rejections.

4.4.2 Para-Boost Multi-Classifier Systems (PB-MCS)

DCS-LA uses K-NN to define local accuracy. There are various possible choices for distance functions that can be used for K-NN. However, it is not necessary that the distance function used would maintain same sense of "nearness" as the underlying individual experts. BKS, on the other hand, tries to learn a probability mass function for all possible outcomes from different experts. Considering a large number of classes and a large number of experts, any non-parametric probability mass function estimation would require invariably large number of data points. Also, as discussed before, the diversity and statistically significant results of individual features motivate us to use vertical decomposition of feature set for MCS. Hence, in Para-Boost MCS, we would

learn a "fuser" classifier or a "meta-classifier", which combines the results of individual systems to make final prediction. Here, as against Adaboost, where sequentially weak classifiers are learned, we boost the classification result combining the results of vertically decomposed features in a parallel fashion; hence the name Para-Boost(PB). We would also like to point out the difference between PB and Stacked Generalization. Stacked Generalization uses horizontal decomposition to form several cross-validation tests. Each learner is trained using set-hold-out strategy where one set is used to generate features for meta-classifier and remaining sets are used to learn individual experts. In PB, we use vertical decomposition and hence each expert is trained on unique feature set using a training set, while a testing set is used to generate meta-classifier. The difference between to approaches can be illustrated by a simple example. Figure 4.1 shows a 2-dimensional data with two classes. Figure 4.2 shows how data is divided into 5-folds for learning a meta-classifier using stacked-generalization, and Figure 4.3 shows the division of data into two feature sets for para-boost model.

Given a feature set, there are numerous ways in which vertical decomposition can be done. There does not exist an optimal decomposition method. However, in literature, it has been suggested that each decomposed group be based on certain physical attributes so that feature dimensions in each group are homogeneous. Keeping this in mind, we propose to use features from different methods (see chapter 2) $M1 - M9$ as already decomposed sets. The idea is that each feature extraction method exploits different characteristics of audio signal. For example, MFCC characterize energy in different Mel-Frequency bands, whereas NBACF explores similar spirit in time domain using autocorrelation coefficients. Hence, MFCC and NBACF can be considered as a valid vertical decomposition.

In PB, classification outputs of individual system will be used as input features for fuser classifier. There are many ways in which we can characterize these input features.

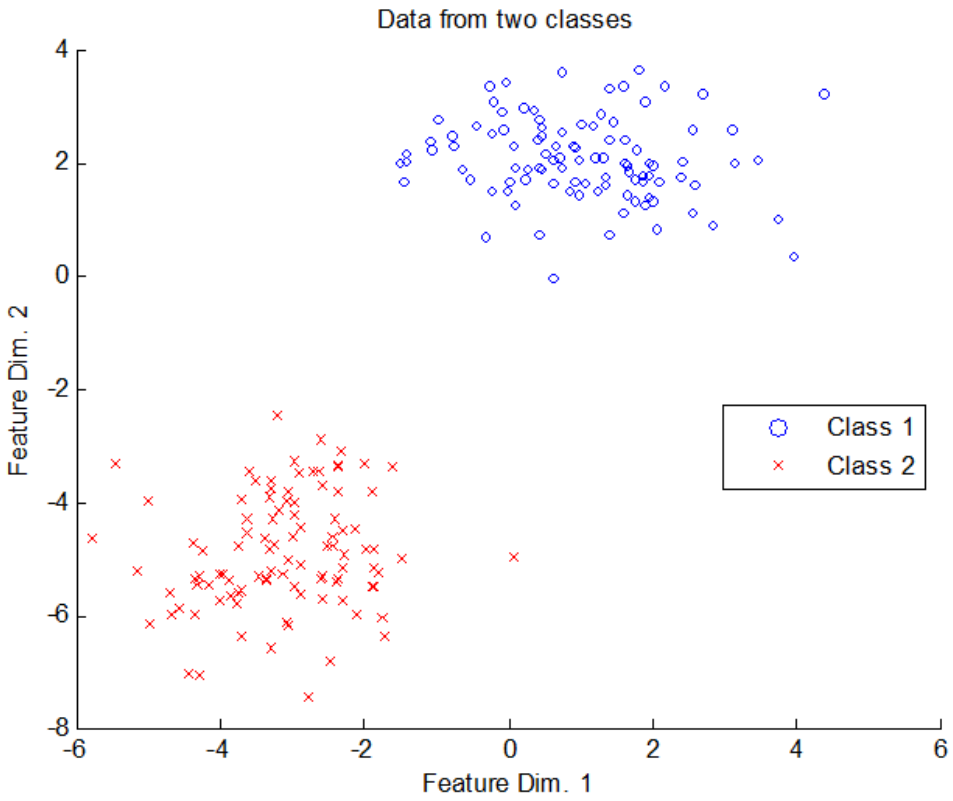

Figure 4.1: Example Dataset with two classes

Also, since this is a multi-classification problem, we can form different architectures for boosting. Based on these distinctions, we propose four variations of PB. We will present each of those in the following discussion.

PB1 - Probability outputs

In PB1, we propose to use classifier fusion system with soft outputs from individual experts. Hence, in this system, we assume that each individual classifier assigns a soft score (akin to probability) corresponding to each of the possible M classes instead of giving one single output:

$$\Psi_i^{PB1} : f_i \rightarrow \mathbb{R}^M \tag{4.11}$$

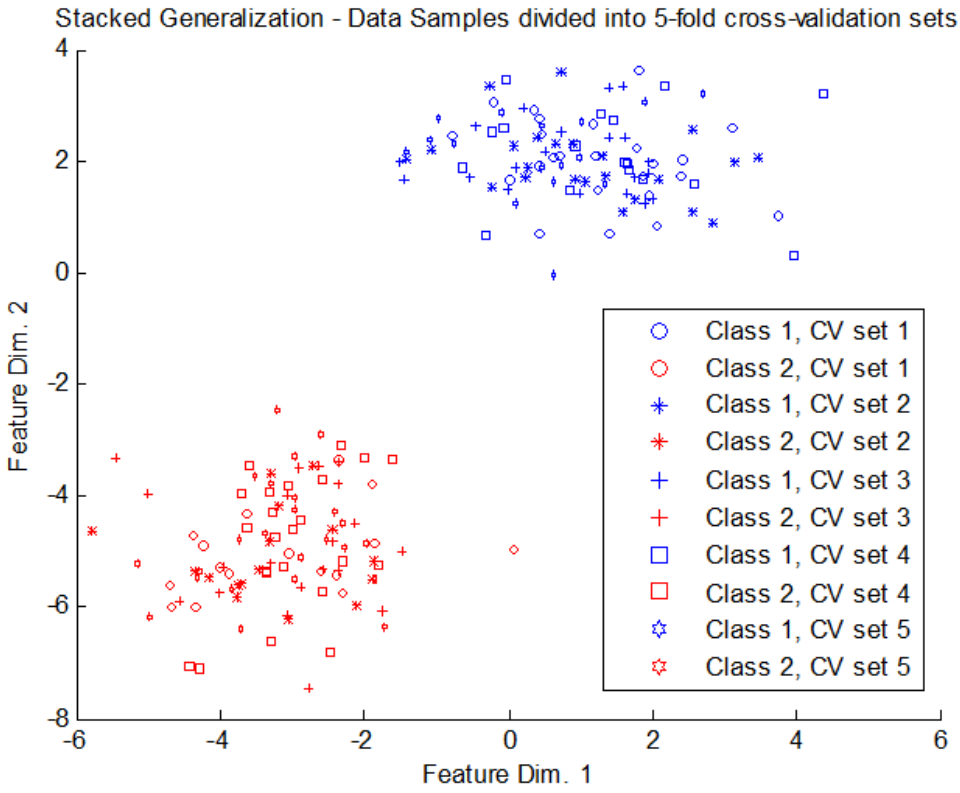

Figure 4.2: Stacked Generalization - Each Cross Validation set is held out once to learn an expert, and the held out set is then used to generate features for meta classifier/fuser

For a given sample t, the output scores from all the classifiers are combined into one vector of R^{MN} dimension which is used as a feature F_{PB1} for the fuser classifier.

$$F_{PB1}(t) = \begin{bmatrix} \Psi_1^{PB1}(t) \\ \vdots \\ \Psi_N^{PB1}(t) \end{bmatrix} \tag{4.12}$$

This feature can be used to train a fuser classifier. We tried several choices for fuser, and initial results favored SVM. Hence, we use SVM for combing the soft outputs to give final prediction. We use LIBSVM library for SVM, wherein one-vs-one classification scheme is used for multi-class classification. Figure 4.4 shows a basic unit of this model:

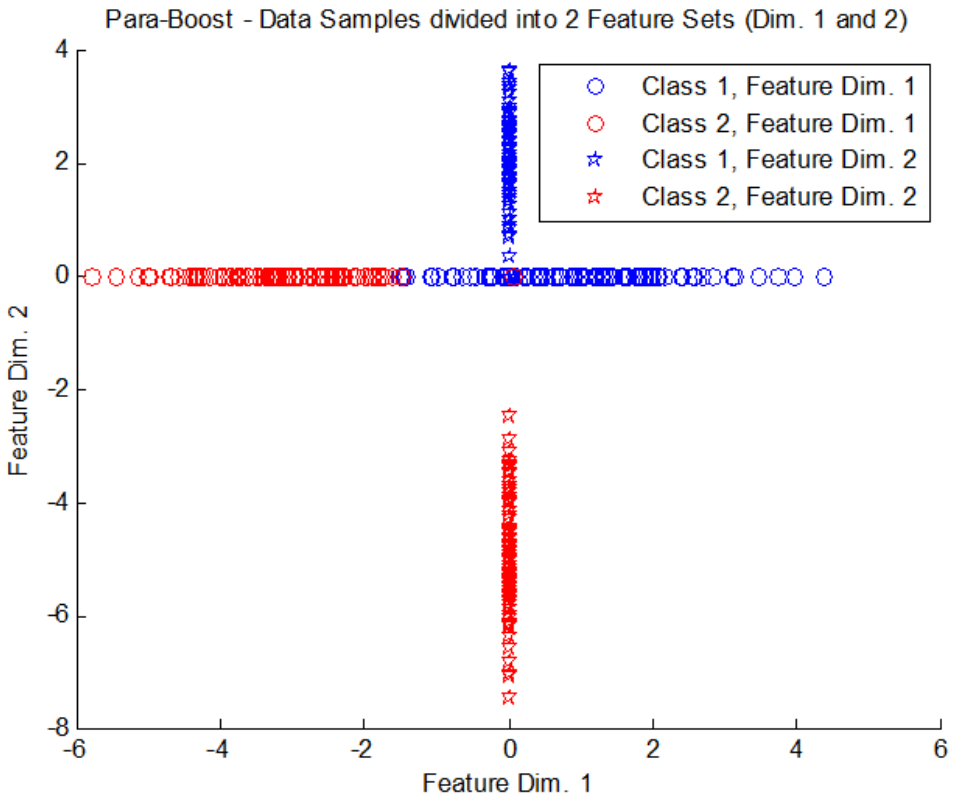

Figure 4.3: Para Boost - Each Feature Set (each dimension in this example) is used to learn an expert. Classification outputs from each feature set are stacked together to generate feature for meta classifier/fuser

Class labels from all vertically decomposed sets is combined to give final prediction, as shown in Figure 4.5.

PB2 - 1-M coding scheme

In previous scheme, we used soft scores from individual experts as input for fuser. However, in general, these scores are correlated for a single classifier. For example, if probability distribution output from individual experts is used, these values will sum to one

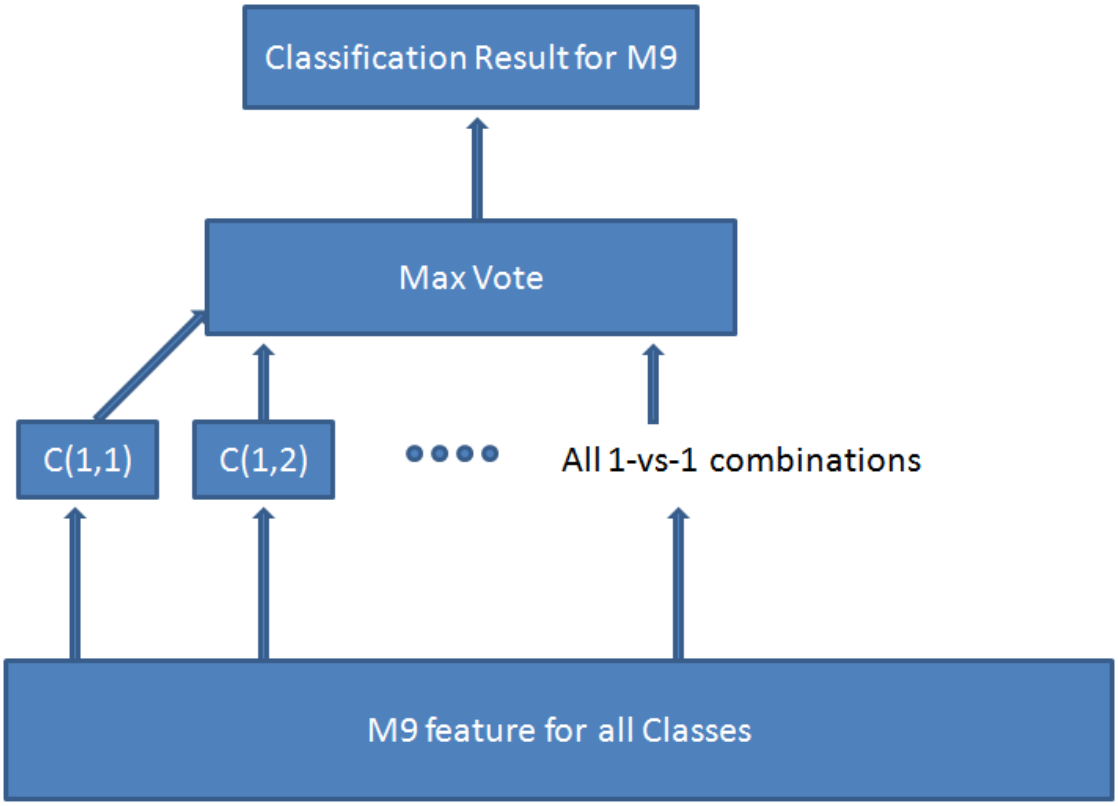

Figure 4.4: Single Expert - Vertically Decomposed Set with SVM classifier

for each expert. Hence, in this variation, PB2, we propose to use 1-M coding scheme. Given a sample t, we can arrange the classifier outputs into 1-M scheme as follows:

$$\Psi_i^{PB2}(t) = \begin{bmatrix} \delta(\Psi_i(t), 1) \\ \vdots \\ \delta(\Psi_i(t), M) \end{bmatrix} \tag{4.13}$$

where $\delta(i, j)$ is as defined in Equation 4.9.

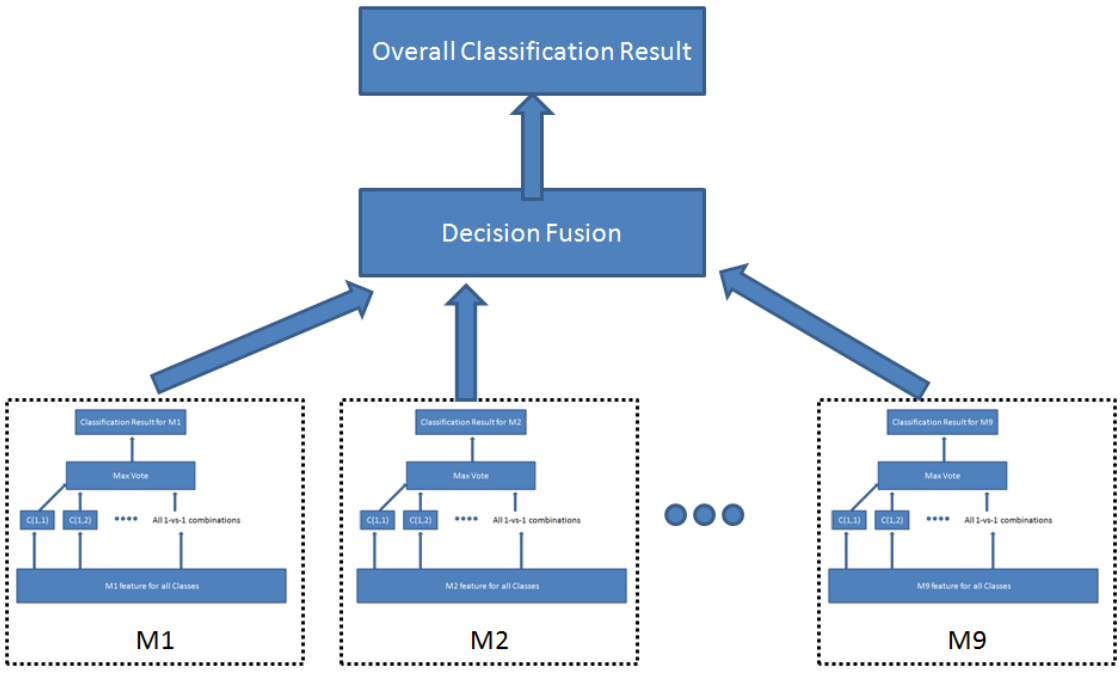

Figure 4.5: Fuser combining the predictions of experts

Thus, given the decisions from all the N classifiers, a new feature vector for PB2 fuser is formed by vertically stacking each of the $1 - M$ coded outputs:

$$F_{PB2}(t) = \begin{bmatrix} \Psi_1^{PB2}(t) \\ \vdots \\ \Psi_N^{PB2}(t) \end{bmatrix} \tag{4.14}$$

Finally, this new feature vector, $F_{PB2}(t)$ is used as input feature for MCS system's final stage fuser.

PB3 - Soft 1-M coding scheme

In PB1, we use correlated output scores as input features for fuser, whereas in PB2, we use 1-M coding scheme. Essentially, in PB2 we assume that the score is 1 for the classifier with maximum score in similar setting in PB1. Thus, in one scheme we muddy

the dominant class with correlated scores for other classes, and in the other scheme we simply ignore the degree of dominance and make it absolute. In PB3, we propose to use the score of the dominant class instead of hard 1-M coding scheme. Thus PB3 seems like a reasonable balance between extremities of PB1 and PB2. Given a sample t, the individual classifier outputs can be arranged using the following to give $\Psi_i^{PB3}(t)$

$$\Psi_i^{PB3}(t) = \Psi_i^{PB2}(t) \circ \Psi_i^{PB1}(t) \tag{4.15}$$

where $\circ$ denotes the Hadamard product. Finally, a new feature vector for PB3 fuser is formed by vertically stacking each of the $\Psi_i^{PB3}(t)$ outputs:

$$F_{PB3}(t) = \begin{bmatrix} \Psi_1^{PB3}(t) \\ \vdots \\ \Psi_N^{PB3}(t) \end{bmatrix} \tag{4.16}$$

PB4 - 1-Vs-1 Scheme

In previous methods, final outputs from all individual feature sets were used to form MCS feature vector. However, it is natural to consider a MCS in which fusion is first performed at individual 1-vs-1 classifier levels followed by max-voting scheme to give final prediction. This MCS system can be explained using the Figure 4.6

4.4.3 Horizontally decomposed Para-Boost (HPB)

Adaboost is based on the horizontal decomposition method where each subsequent weak classifier focuses on creating error diversity by sampling incorrectly predicted data points more often than the others. Vertical decomposition, on the other hand, creates diversity by first allowing coherent decisions to be made on individual feature subspace, and finally combine decisions from various sub-spaces. Hence, it is natural to

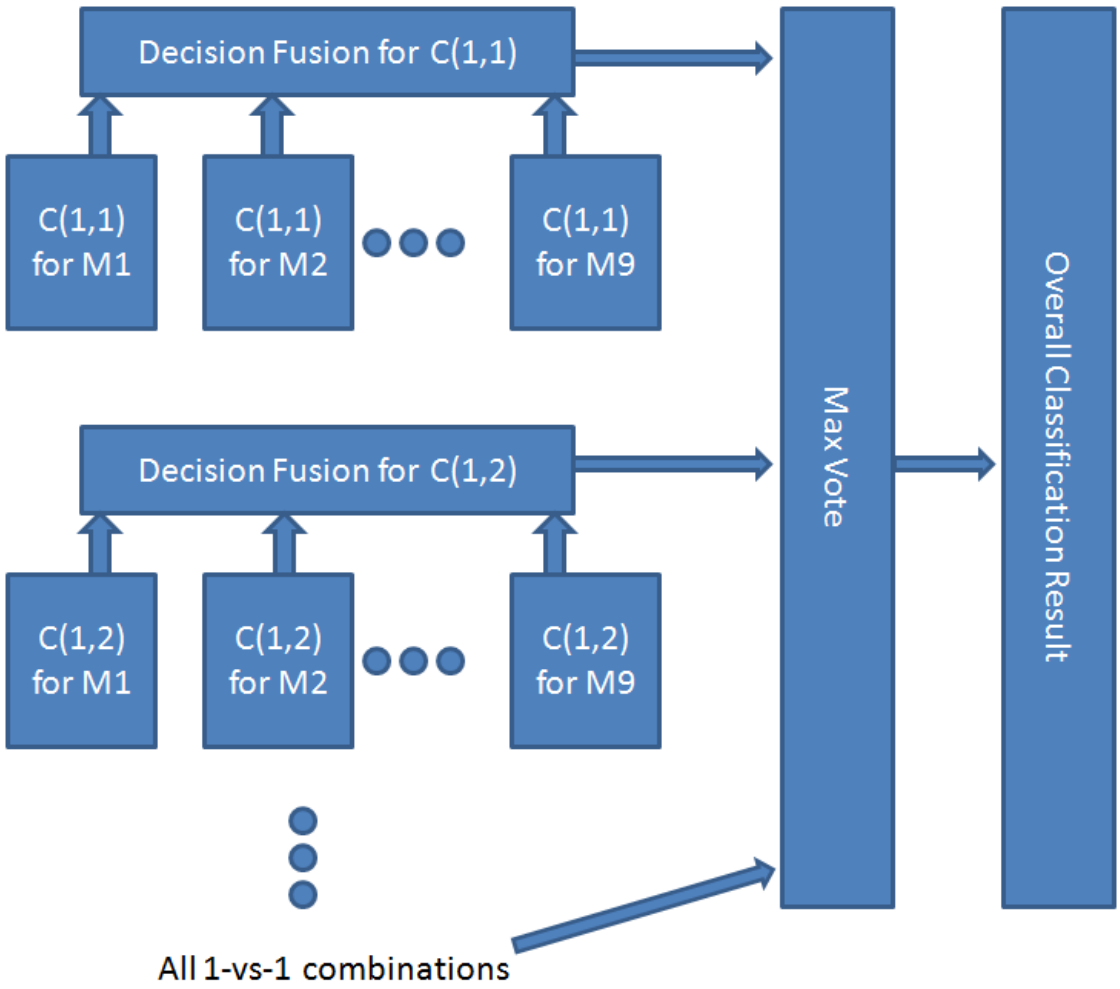

Figure 4.6: 1-Vs-1 Para-Boost Model - PB4

combine the two methods to improve the overall performance of the system. Hence, in this approach, we first perform Adaboost on each of the feature set, and then do a final SVM fusion based on predictions of Adaboost stage.

4.4.4 Grouping Based Para-Boost MCS (GPB)

In Section 4.6, we will see that the performance of PB and its variants is bottlenecked to about 88%. This is due to the fact that we have exploited the available diversity

in the features. Hence, in order to push the results beyond PB, we need to introduce more diversity, without increasing more feature sets. One way to do so would be to use context-dependent PB systems where each PB system strives to optimally exploit the diversity in samples native to its context.

In order to form context-dependent PB system, we propose to first group the training data into two groups using supervised clustering. However, grouping of raw data would not be helpful. Hence, we perform the grouping based on the best performing feature set, among all the features. Then, we learn PB system for each of the two groups. Thus, GPB can be describe as follows

1. Group all the training data using K-Means clustering and best performing feature set, $\hat{f} = \arg_i \max Acc(f_i)$, wehre $Acc(f_i)$ is the average classification accuracy for feature set f_i.

2. Learn context-dependent classifiers for each group again.

3. Learn context-dependent PB classifier for each group.

4. Given a test data, use PB classifier (learned in previous step) corresponding to its group membership.

4.5 Forward Sequential Search for Para-Boost MCS

In general, one can have a large number of vertically decomposed features sets at his disposal. Then one begs the question: Should all the feature sets be used? We try to answer this question by suggesting a way to sequentially add feature sets in PB-MCS system.

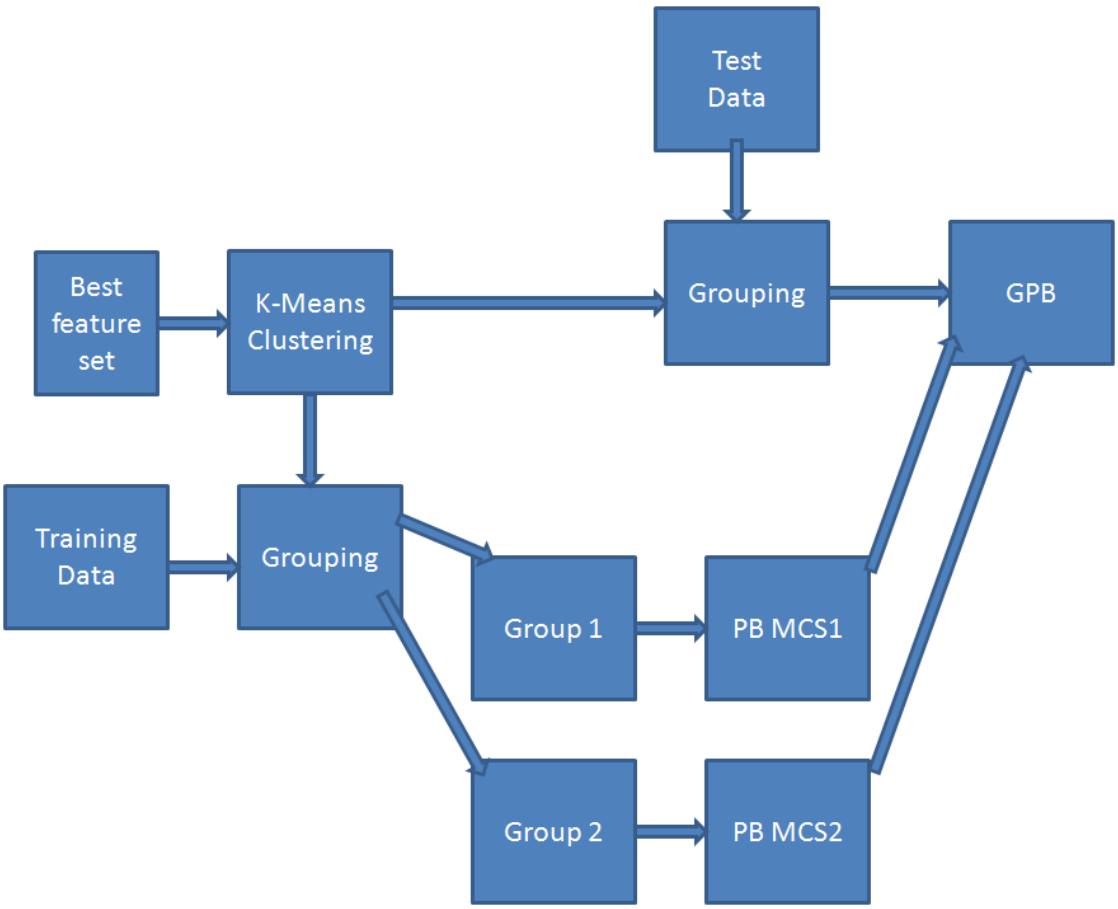

Figure 4.7: Grouping Based Para-Boost Model (GBP)

It should be noted that diversity is very crucial for a successful MCS. Hence, at any given step, one can check which feature set among the options available, provides maximum diversity with current PBS system. Using this ideology, we propose the following forward sequential approach to build PB-MCS.

1. Learn individual classifier Ψ_i for each feature set f_i

2. Let $\boldsymbol{\Psi}_{\boldsymbol{FS}}^{\boldsymbol{n}}$ be the set of selected classifiers for PB at iteration n. Initialize it as:

$$\boldsymbol{\Psi}_{\boldsymbol{FS}}^{\boldsymbol{0}} = \{\Psi_{i_0}\} \ni i_0 = \arg_i \max \sum_{\forall t} \delta(L(t), \Psi_i(t)) \tag{4.17}$$

Also, let $\boldsymbol{\Psi_{rem}^{n}}$ be the set of remaining unselected classifiers at iteration n. Initialize it as: $\boldsymbol{\Psi_{rem}^{0}} = \{\Psi_1, \ldots, \Psi_N\} \setminus \boldsymbol{\Psi_{FS}^{0}}$

3. At each iteration n

   (a) Learn PB classifier, $\boldsymbol{\Psi_{PB}^{n}}$, at current iteration using experts from $\boldsymbol{\Psi_{FS}^{n-1}}$

   (b) Find class-wise Accuracy for each pair of class and classifier:

$$Acc_m^n(f_i, \Psi_i) = \sum_{t:L(t)=m} \delta(L(t), \Psi_i(t)) \tag{4.18}$$

   Here, $Acc_m^n(f_i, \Psi_i)$ is accuracy of classifier Ψ_i corresponding to feature set f_i and class m. Similarly, let $Acc_m^n(F_{PB}, \boldsymbol{\Psi_{PB}^{n}})$ be accuracy of PB classifier for class m.

   (c) Find the most diverse classifier Ψ_{i_n} at current iteration as follows:

$$\Psi_{i_n} = \arg_i \max \sum_{\forall m} \mathbb{I}(Acc_m^n(f_i, \Psi_i) > Acc_m^n(F_{PB}, \boldsymbol{\Psi_{PB}^{n}})) \tag{4.19}$$

   where

$$\mathbb{I}(A) = \begin{cases} 1, & \text{if event } A \text{ is True} \\ 0, & \text{otherwise} \end{cases} \tag{4.20}$$

   (d) Update $\boldsymbol{\Psi_{FS}^{n}}$ and $\boldsymbol{\Psi_{rem}^{n}}$ as follows:

$$\begin{aligned} \boldsymbol{\Psi_{FS}^{n}} &= \boldsymbol{\Psi_{FS}^{n-1}} \cup \{\Psi_{i_n}\} \\ \boldsymbol{\Psi_{rem}^{n}} &= \boldsymbol{\Psi_{rem}^{n-1}} \setminus \{\Psi_{i_n}\} \end{aligned} \tag{4.21}$$

   (e) Stop if $\boldsymbol{\Psi_{rem}^{n}} = \varnothing$ or no diverse classifier is found in step 3c

Here we chose to use class-wise diversity as a measure as against overall diversity because we believe that the best improvement can be achieved if class boundaries of current PB classifier's are not challenged for classes it is already doing the best.

4.6 Experimental Results and Discussion

4.6.1 Experimental Set-up and Database

We investigate the performance of proposed Para-Boost methods on the ESR database as described in Section 3.3.1. We choose all the 37 classes from Table 3.2. The set-up is also similar to the one described in Section 3.3.1. For individual experts, we selected M1-M9 from Table 3.3. Where applicable, we use sub-framing based method (see Section 2.1) and take average of feature vectors over sub-frames to obtain a single feature vector to represent one data sample.

For a fair comparison, we conducted 30 trials of each MCS system. For each trial, we randomly sampled up-to a maximum of 400 data points for each class. Then, we randomly divided these samples into training and testing sets with 70% and 30% proportions, respectively. In total, we end up with 10103 training samples and 4331 testing samples.

First, using the training data, we learn models for each of the 9 experts. Next, the output labels (and/or scores) are used to generate training vectors for the PB classifier, leading to learning of the fuser. For a test sample, output labels (and/or scores) from trained individual experts are combined in a similar fashion as it was done for training data. Finally, fuser's output label is used as the final classification guess. For fuser, we tested three classifiers - K-NN, GMM and SVM. The performance of KNN and GMM was very poor, as compared to SVM. Hence, we decided to use SVM as final fuser. For SVM, we used LIBSVM package [6].

For evaluation of the methods, we used Weighted Classification Accuracy (WCR) for n^{th} trail defined as follows:

$$WCR_n(\Psi_{PB}) = \frac{1}{M} \sum_{\forall m \in \mathbb{M}} \frac{\sum_{t:L(t)=m} \delta(L(t), \Psi_{PB}(t))}{|\{t : L(t) = m\}|} \tag{4.22}$$

This is done in order to remove bias from classes with more number of test samples. Finally, Mean of WCR over 30 trials is used as final measure to evaluate the performance of a MCS system:

$$ACR(\Psi_{PB}) = \frac{1}{30} \sum_{n=1}^{30} WCR_n(\Psi_{PB}) \tag{4.23}$$

4.6.2 Results and Discussion

As described in Section 4.4.1, we use DCS-LA and BKS for benchmarking performance of proposed Para-Boost models. Figure 4.8 shows how overall classification accuracy of DCS-LA changes with choice of K as defined in Equation 4.8

It can be seen that the best performance is achieved for $K = 3$. However, accuracy for $K = 5$ is not to far behind. Hence, with hopes of a robust classifier, we decide to use $K = 5$ for all trails when using DCS-LA as a benchmark.

In Figure 4.9 we compare the performance of the two benchmark methods, and those of the proposed methods. Considering the Mean of WCR, $ACR(\Psi_{PB})$,it is clear that the all the proposed methods show significant improvement over the performance of benchmark methods. In particular, BKS performs worst amongst all. This should not be surprising as BKS uses empirical probability mass function as a tool to model behavior of experts in a high dimensional space with $M^N = 37^9 \approx 10^{14.11}$ possible discrete values, with a small number of data samples, i.e. 10103 training samples. On the other

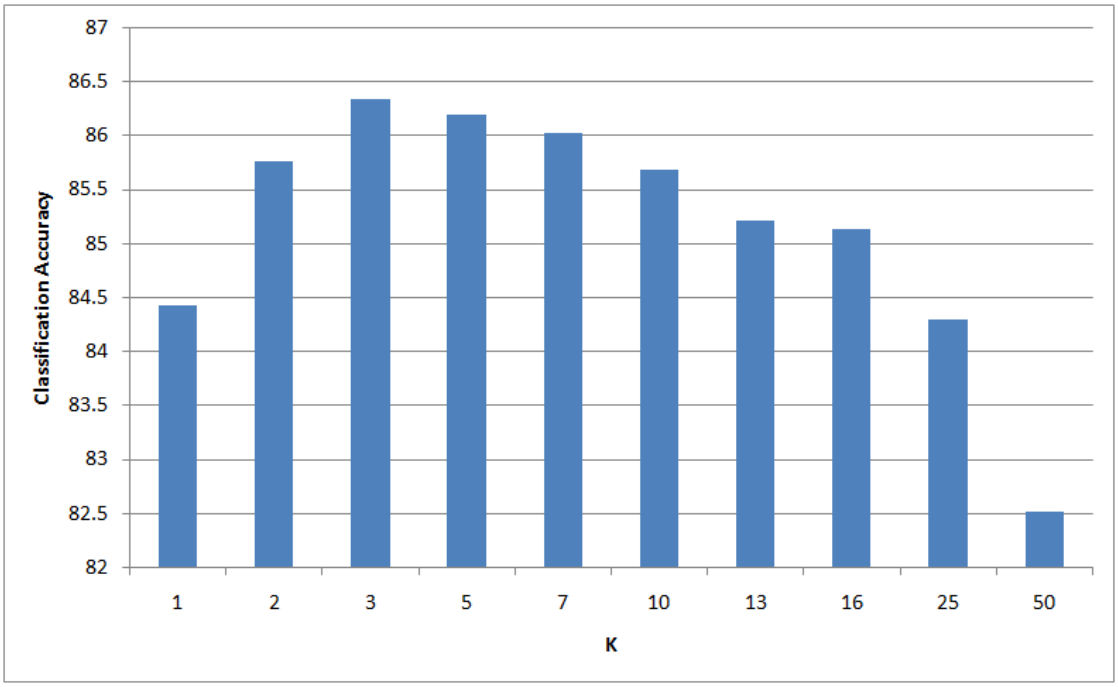

Figure 4.8: Effect of K in Performance of DCS-LA

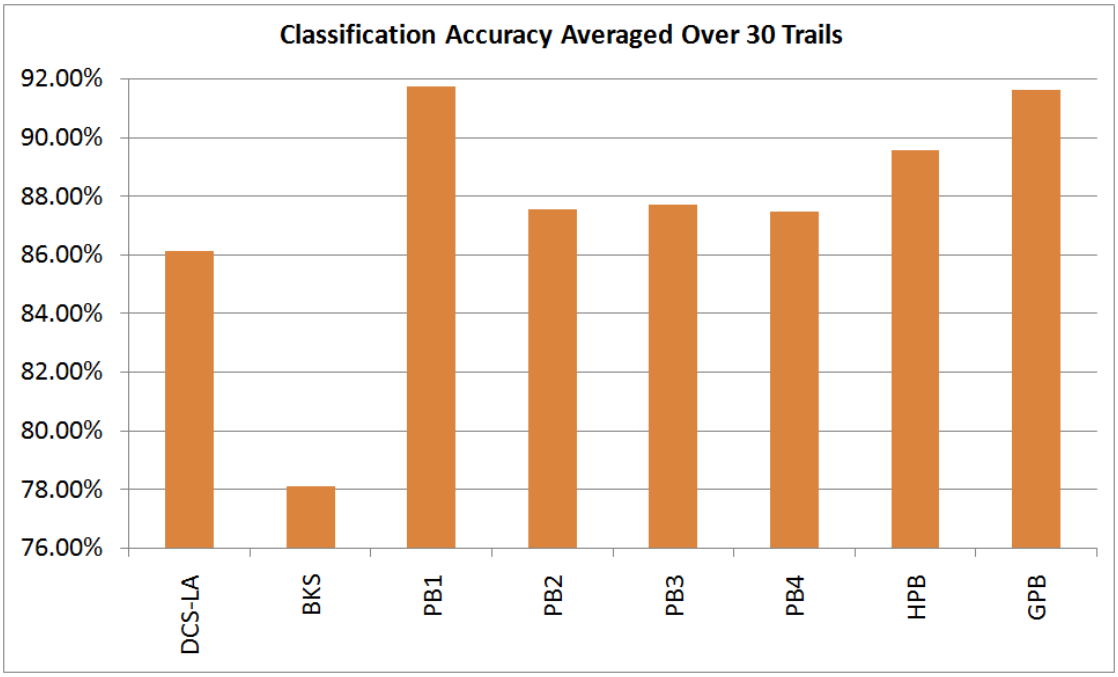

Figure 4.9: Average Classification Accuracy Over 30 Trials

hand, PB methods use SVM to learn the SVs, as against learning parametric or non-parametric probabilistic model. DCS-LA also does not learn any probabilistic model. It just ranks the experts in local regions based on their performance in that region.

Amongst the six proposed methods, PB1 and GPB perform the best, followed by HPB. Performance of PB2, PB3 and PB4 is very similar to each other, and is slightly better than that of DCS-LA. In order to gain more insight into performance, we show the results for all the 30 trials in Figure 4.10

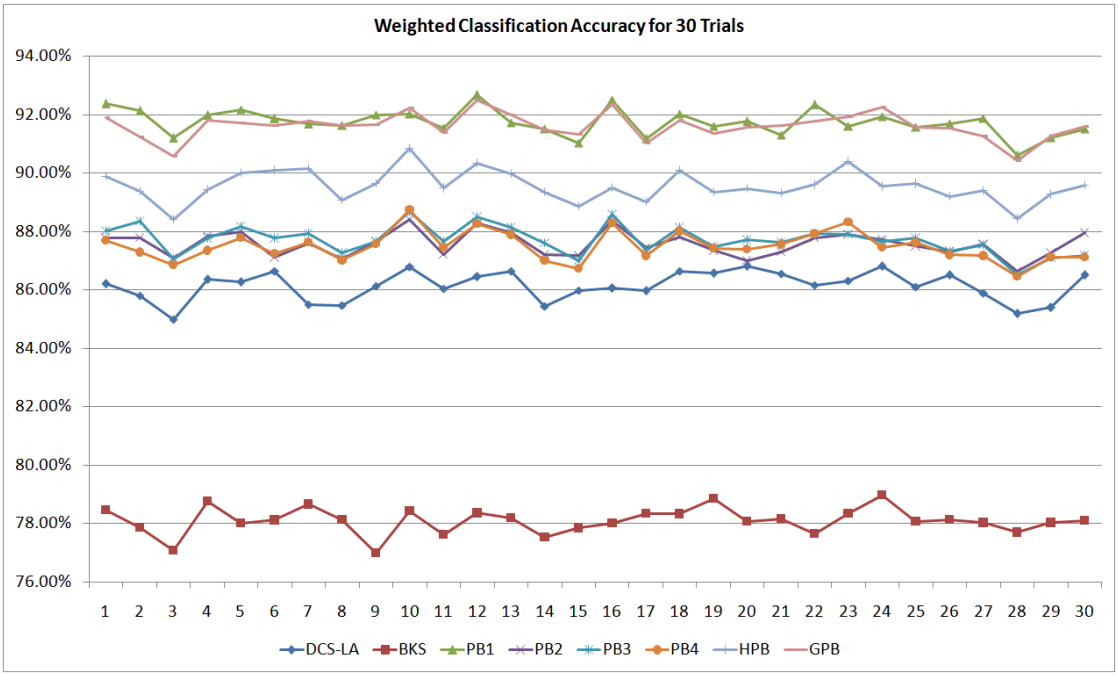

Figure 4.10: Weighted Classification Accuracy for 30 Trials

In order to verify the statistical significance of classification results, we also performed two tests, McNemars test and paired t-test. For more details on the tests, please refer to Section 3.3.3. Table 4.5 and Table 4.6 show McNemar's test statistic for one trial and paired t-test statistic for 30 trials, respectively. Recall that the null hypothesis, that two methods under consideration have same error rate, is rejected if M_{stat} is greater than $\chi^2_{1,0.9999} = 15.1367$ for McNemar's test, and it is rejected if absolute value of T_{stat}

Table 4.5: McNemar's Test Statistic for 1 trial

	BKS	PB1	PB2	PB3	PB4	HPB	GPB
DCS-LA	186.37	3017.28	2677.44	2697.49	2918.25	2913.79	3174.12
BKS		2695.28	2367.30	2388.47	2516.73	2504.11	2744.30
PB1			126.06	137.39	58.07	31.00	1.70
PB2				0.35	1.79	41.16	36.20
PB3					0.10	32.90	32.58
PB4						37.99	36.40
HPB							37.07

Table 4.6: Paired T-Test Statistic for 30 trials

	BKS	PB1	PB2	PB3	PB4	HPB	GPB
DCS-LA	93.64	-61.00	-15.93	-17.02	-15.48	-40.00	-73.62
BKS		-132.08	-101.18	-92.49	-91.83	-115.73	-159.81
PB1			71.99	82.10	61.68	27.78	2.36
PB2				-2.73	1.53	-28.63	-72.52
PB3					4.90	-29.90	-66.74
PB4						-39.13	-74.43
HPB							-33.41

is greater than $t_{29,0.99995} = 4.5305$; both for a p-value of 0.0001. See Equation 3.4 and Equation 3.5 for definition of M_{stat} and T_{stat}, respectively. We will use these test results to draw meaningful conclusions in the following discussions.

Comparison of coding schemes in PB1-PB4

According to both the tests, PB1 gives significantly better results than the others. The only difference between PB1-PB4 is how the feature for meta-classifier is coded from labels/scores of individual experts. Yet, the performance difference between PB1 and the rest is considerable. This could be because, even though there is correlation between

scores from one expert, the nature of correlation is complicated. In other models, we either use 1 or a soft score, and ignore the scores for other classes, thereby reducing the dimensionality of vector from $Mv1$ to 1×1, thereby losing considerable information. Hence, even though correlation exists between score of one expert, they seem to be useful when used with those of other experts. This can be particularly helpful when the difference between largest and second largest scores is not much. During such cases distribution of scores over remaining classes can prove to be discriminatory. PB2, PB3 and PB4, on the other hand, give very similar results, and all 3 pairs fail the McNemar's test.

Use of Horizontal and Vertical Decomposition together

Though HPB does not perform as well as PB1 and GBP, it shows overall improvement with respect to other PB methods. This can also be verified from the statistical significance tests. Figure 4.11 compares the performance of SVM and AdaBoost classifiers for all 9 experts. Clearly, AdaBoost performs poorly for individual experts. However, using PB with AdaBoost gives significant improvement over individual experts. This can be explained by the fact that even though AdaBoost gives low accuracies for individual experts, the results from different experts are pretty diverse. This can be confirmed from Correlation Diversity measure for AdaBoost as shown in Table4.7. When compared to correlation diversity measure for SVM in Table 4.2, SVM on an average has slightly lower correlation (≈ 0.032), meaning SVM is more diverse. However, despite individual accuracies being low for AdaBoost, it shows sufficient diversity. This also explains why PB1 performs slightly better that HPB.

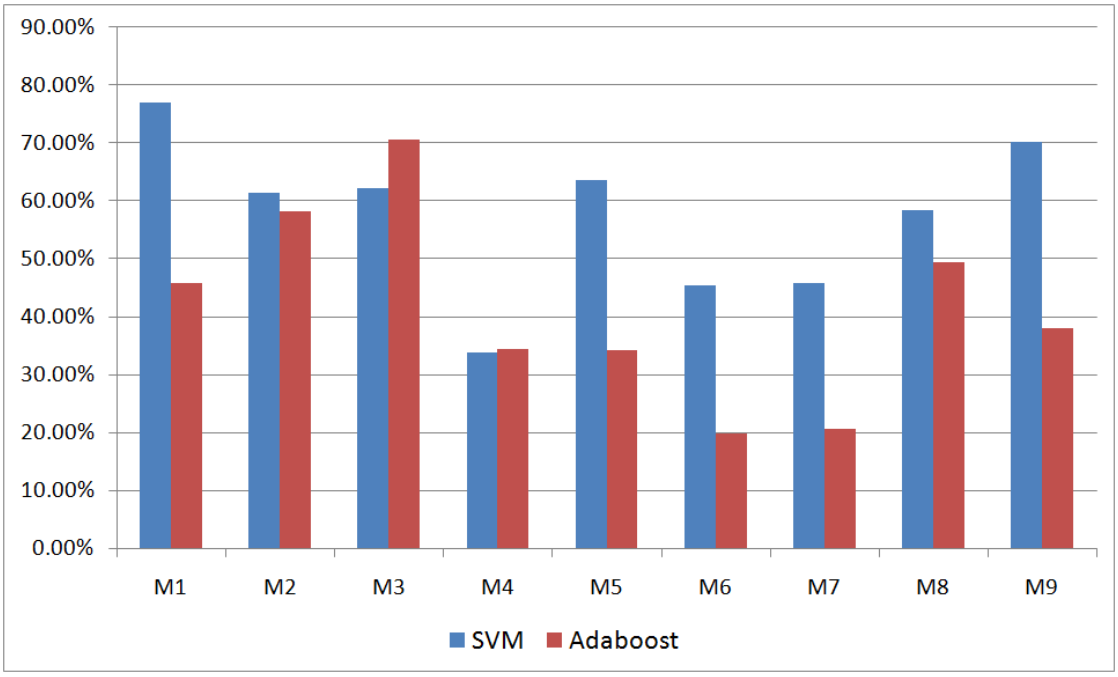

Figure 4.11: Performance Comparison for SVM and AdaBoost

GPB Vs PB1

Results for both GBP and PB1 are very similar for all the 30 trials. In fact, both McNemar's test and T-test fail to reject the null hypothesis they have same error rate for a p-value of 0.0001! Also, computational complexity is higher for PBS as it is a two step process. Hence we conclude that even though GPB has good performance accuracy, PB1 should be favored over it.

4.7 Conclusion

In this chapter, we first introduced Multi-Classifier Systems. We presented the differences between feature fusion and MCS. We also discussed different types of MCS, and role of diversity in MCS. Significant differences between Horizontal and Vertical Decomposition approaches of MCS were also discussed.

Table 4.7: Correlation Diversity Measure for AdaBoost

	M1	M2	M3	M4	M5	M6	M7	M8	M9
M1	1.000	0.115	0.242	0.232	0.296	0.257	0.180	0.130	0.427
M2	0.115	1.000	0.125	0.118	0.051	0.095	0.076	0.047	0.008
M3	0.242	0.125	1.000	0.259	0.251	0.103	0.079	0.153	0.173
M4	0.232	0.118	0.259	1.000	0.393	0.054	0.154	0.135	0.177
M5	0.296	0.051	0.251	0.393	1.000	0.135	0.153	0.094	0.238
M6	0.257	0.095	0.103	0.054	0.135	1.000	0.251	0.114	0.275
M7	0.180	0.076	0.079	0.154	0.153	0.251	1.000	0.087	0.185
M8	0.130	0.047	0.153	0.135	0.094	0.114	0.087	1.000	0.132
M9	0.427	0.008	0.173	0.177	0.238	0.275	0.185	0.132	1.000

We then show that there is sufficient diversity in vertically decomposed experts consisting of different feature sets. Motivated by this, we presented Para-Boost MCS, which combines the ideas of Stacked Generalization and Random Subspace methods. In total, we proposed 4 different variations of PB based on how features we coded for meta-classifier. We reported that one of the coding schemes, PB1, which uses all the score outputs from individual experts, outperforms other coding schemes. We then proposed two more variations - HPB and GPB. Both HPB and GBP provide significant improvement over individual experts and also perform better than benchmark methods - DCS-LA and BKS.

Overall, PB1 and GPB give best results. We need further analysis to understand why GPB does not outperform PB1. Also, further analysis is needed to see how we can further improve the performance of Multi-Classifier Systems.

Chapter 5

Content Based Environmental Sound Retrieval

5.1 Introduction

Multimedia available on Internet has grown exponentially over the last decade. With popularity of social networks, user captured data is also publicly available for download. However, it has become harder to find the right kind of data on the Internet. Hence, it is natural to develop mechanisms which help in retrieving relevant data. A lot of research has been done for Music Information Retrieval. On the other hand, environmental sounds have not received as much attention despite the fact that the availability of such sounds has increased tremendously. Hence, in this chapter, we investigate and formulate an environmental sound retrieval system.

In particular, we are interested in query-by-example retrieval system. In this system, given a query, the task is to retrieve all documents in the database which are relevant to the query. Also, the retrieved relevant documents are ranked by "how close these sound to the query?". Retrieval systems can be broadly classified into two categories: Context-based and Content-based retrieval systems. Context-based retrieval systems use semantic information such as tags, file-names, etc. to retrieve relevant documents[39, 27, 25, 30]. Content based systems, on the other hand, use the content of the query audio itself for retrieval. Context based systems rely on information usually added by users. For example, a sound clip of Machine Gun could be tagged with multiple tags, such as

machine guns, violence, guns, war, crime, etc. There is a huge semantic gap between the tags and the content of the audio. Hence, there are not of works which try to bridge this gap by linking the content to the tags[26]. However, this is still largely an open problem. In this work we focus on developing content based retrieval system where we would match the ”content” of the query to that of all the documents/audio clips in the database.

5.2 Related Work

In terms of general audio retrieval, not a lot of work as been done. In [31, 21, 2], authors study the problem of recognition and retrieval for animal sounds. Sport sounds have been studied in [40, 1], and birds sounds have been modeled for recognition and retrieval in [4, 28].

Some of the earliest work on general audio retrieval includes that by Foote[16] and Zhang and Kuo[64]. In [16], author uses MFCC features to create quantized dictionary of features, which is then used to represent an audio as a histogram over the dictionary. However, the data set chosen for retrieval is consists of just six classes and is very small. An interesting approach was proposed in [64], where a template of a query consists of parameters of HMM trained over the query clip. Ranking and retrieval was done based on $P(d/\lambda_q)$ where d is a sample in database, q in a query, and λ_q are parameters of HMM for query q. In a more recent work[58], Wichern *et al.* also use HMM to model the trajectory of features to form a template but in a more sophisticated manner. The first select control points in feature frames, then model the trajectory of these control points instead of modeling all the frames. This allows them to first align query and a database sample, and then estimate the probability $P(d/\lambda_q)$. They assume that the database is completely unlabeled. A fast indexing framework is also proposed by using

spectral clustering to cluster the unlabeled data. HMM has been very successful in field of music and speech retrieval. There are more structured sounds and consist of smaller units or characteristic properties which are consistent over samples. However, the same does not hold true for environmental sounds. There can be vast variation between two audio clips of same category. For example, sound of airplane flying by can be that of a midair, takeoff or landing. The evolution of the features would vary a lot within same category of class. Time-warped matching of control points is not capable of encapsulating all possibilities. Moreover, in all the previous works, the number of classes and the database itself is small.

5.3 Problem Formulation

Traditionally, retrieval problem can be formulated in two ways:

- Fully Labeled Data Set: In this case, it is assumed that the data set is fully labeled into pre-defined categories. The query is expected to retrieve all the documents in the database from its corresponding query. Hence, machine learning approaches are used to learn classification model for the underlying categories and predicted label is used to retrieve all the relevant samples.

- Unlabeled Data Set: In this case, there is no assumption of pre-defined categories. The goal is to create an indexed data set based on a query model. Given a query, a retrieval process based on indexing is performed to output a ranked list of documents which are most similar to the query.

Today, we have a huge amount of environmental sound data on the Internet. These samples can be labeled or unlabeled. For example, a lot of sounds on freesound.org, a repository for user submitted sounds, are tagged. On the other hand, a lot of these sounds

are incorrectly labeled or not labeled. With the advent of crowd-sourcing models, it is more likely to have such partially labeled data sets. Hence, it is natural to model a retrieval framework which appropriates this feature of data-sets instead of formulating a harder retrieval framework of an unlabeled dataset. Hence, we formulate out retrieval problem as follows:

- Data-Set: It is assumed that the data-set $D = \{D_L \cup D_U\}$ is partially labeled with the help of experts. It is assumed that underlying classes/categories $C = \{C_1, \ldots, C_N\}$ are mutually exclusive of each other.
- Query: Given a query q with label C_q, all the documents from dataset D are retrieved such that both labeled and unlabeled documents from class C_q are ranked higher that other irrelevant documents.

5.4 Proposed Method

An overview of the proposed retrieval framework is shown in Figure 5.1 As shown in

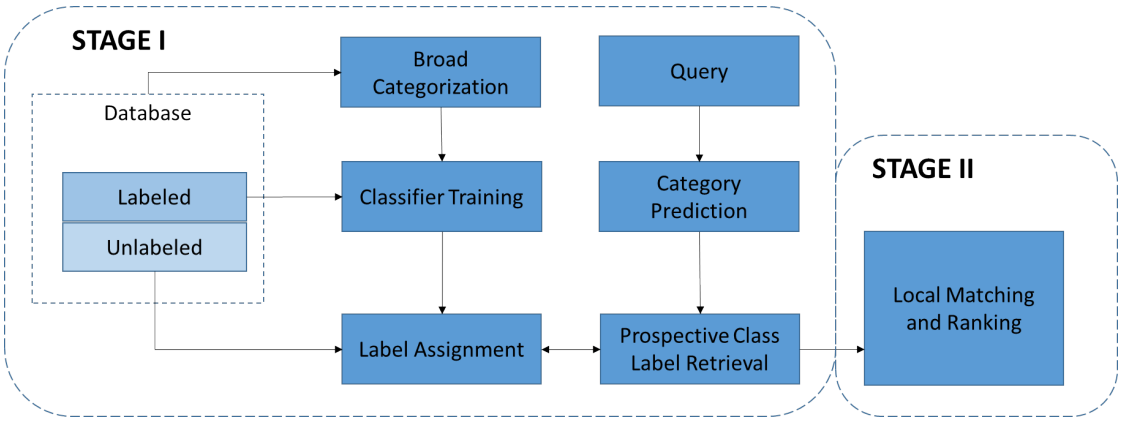

Figure 5.1: Block Diagram of Proposed Framework

the block diagram, the system is divided into two stages. Stage I of the system is an off-line phase and consists of following steps:

1. **Broad Categorization:** In this step, the samples in the database are categorized into three categories based on their signal characteristics: Time Localized Signals, Frequency Localized Signals and Others.

2. **Classifier Training:** Using the labeled data D_L and a set of features, a random forest classifier is trained to predict the labels for the unlabeled data D_U.The classifier training process is iterative wherein the each iterative step uses samples from D_U which are predicted with high confidence, along with labeled data D_L, to retrain the classifier. This process is done separately for the three categories from step 1

Stage II is an on-line phase and is activated when a query comes in. Given a query, following steps are performed in Stage II of the retrieval system:

1. **Prospective Class Label Retrieval:** First, the query is categorized into one of the three categories: Time Localized, Frequency Localized and Other. Then, classifier for that category is used to assign scores to each class. Finally, top few perspective class labels are deemed as "relevant" for the query.

2. **Local Matching and Ranking:** Finally, given the relevant class labels, all the corresponding samples from the dataset D are retrieved and ranked based on a novel signal dependent bag-of-word representation. Rank fusion and diffusion process can be applied to enhance the retrieval performance.

We will now describe the details of each Stage in the following sections.

5.4.1 Stage I

This off-line Stage is divided into following two steps:

Broad Categorization

The dataset, which will be introduced in 5.5.1, consists of 37 environmental sound classes. Some of these classes share certain signal characteristics which can be used to make a broad categorization. The question, one might ask, is "What is the need of such a categorization?" To understand this, let us first see some of the properties. Figure 5.2 shows a few samples from classes Ceramic Collision, GasJetting and Wood-Collision. The top row of figures shows the signal in time domain, while the bottom row

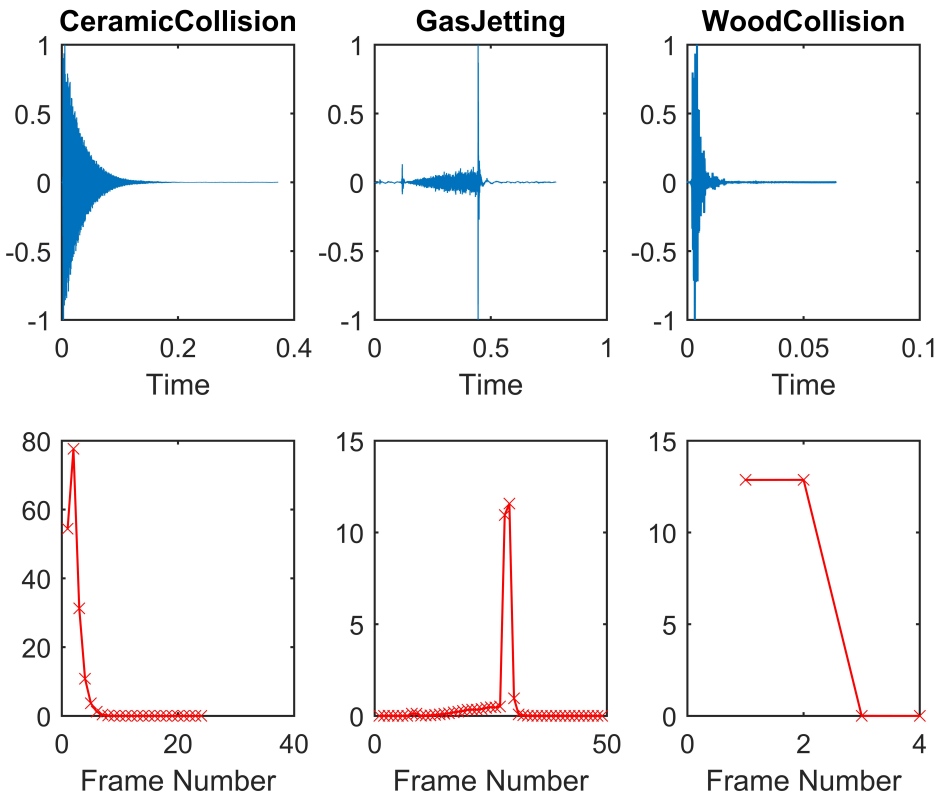

Figure 5.2: Time Localized Signals

shows the smoothened short-time-energy. It can be seen that these sound classes consists of signals highly localized in time, or in other words, are short-lived sound events. In fact, all the samples from these class exhibit this property. Owing to this fact, it is

natural to categorize all these sounds as a separate category which will in turn help in reducing model complexity for the next step of classifier training and thereby help in reducing the "relevant" candidate pool for final local matching and ranking. Moreover, due to the short-lived characteristics of these signals, local matching and ranking would be challenging when matched with classes where temporal information is varying over a long period of time. Finally, it should be noted that classifier training would also benefit from this as we can use a specialized feature that exploits the signal characteristics for this category. Since this property is shared by all the samples from this classes, this categorization can be formulated by a simple approach:

- Normalized Short Time Energy is calculated for an audio sample x by dividing it into frames $x_1, \ldots, x_n$ with 50% overlap.

$$STE = \{ste_i \forall i \in [1, N] \ni ste_i = \frac{||x_i||_2^2}{\sum_{k=1}^{N} ||x_k||_2^2}\}$$

- Smooth the STE curve by a Moving Average Filter and find the peaks in STE:

$$Peaks = \{i : ste_i - ste_{i-1} > 0 > ste_{i+1} - ste_i\}$$

- Merge Peaks which are close to each other, and find the larges peak, ste_L.

- Categorize the signal as time localized if

$$\sum_{i=L-K}^{L+K} ste_i \leq Th_t$$

Hence, by simply finding if the energy is concentrated in a small window, we can find the time localized signals.

Now let's examine another property of signals - energy distribution in frequency domain. Figure 5.3 shows samples from classes Bees, Crickets, and MachineGuns. The

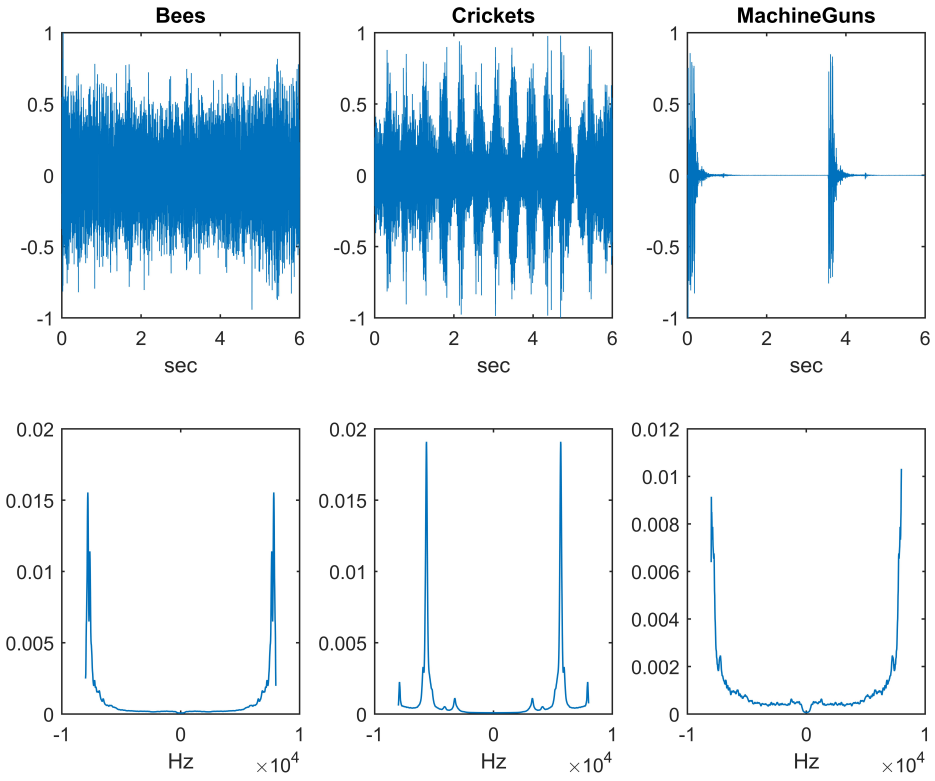

Figure 5.3: Frequency Localized Signals

top row shows the signals in time domain, whereas the bottom row shows the averaged frequency response over frames of the signal. It can be seen that these signals have high concentration of energy at certain frequency locations. Just like time localized signals, categorizing these into a separate category would help in reducing model complexity and allow us to use features that exploit this signal characteristic. This categorization also can be done using similar processing as time localized signals:

- First Short Time Fourier Transform is calculated for an audio sample x by dividing it into frames $x_1, \ldots, x_n$ with 50% overlap, and can be denoted by $\{\text{fft}(x_1), \ldots, \text{fft}(x_n)\}$

- Then we calculate the average frequency response $AVR = \{avr_1, \ldots, avr_{N_{fft}}\} = \frac{1}{n}\sum_1^n \text{abs}[\text{fft}(x_i)]$ and normalized average frequency response $NAFR$

$$NAFR = \{nafr_i \forall i \in [1, N_{fft}] \ni nafr_i = \frac{||avr_i||_2^2}{\sum_{k=1}^{N_{fft}} ||avr_k||_2^2}\}$$

- Smooth the $NAFR$ curve by a Moving Average Filter and find the peaks in $NAFR$ for positve frequencies only:

$$Peaks = \{i : nafr_i - nafr_{i-1} > 0 > nafr_{i+1} - nafr_i\}$$

- Merge Peaks which are close to each other, and find the larges peak, $nafr_L$.

- Categorize the signal as frequency localized if

$$\sum_{i=L-K}^{L+K} nafr_i \leq Th_f$$

Some signals which have been characterized as time localized also exhibit localization in frequency property. One such example is shown in Figure 5.4. Hence frequency localized signal categorization must be made after time localized categorization is done. Finally the remaining signals are categorized as Others category.

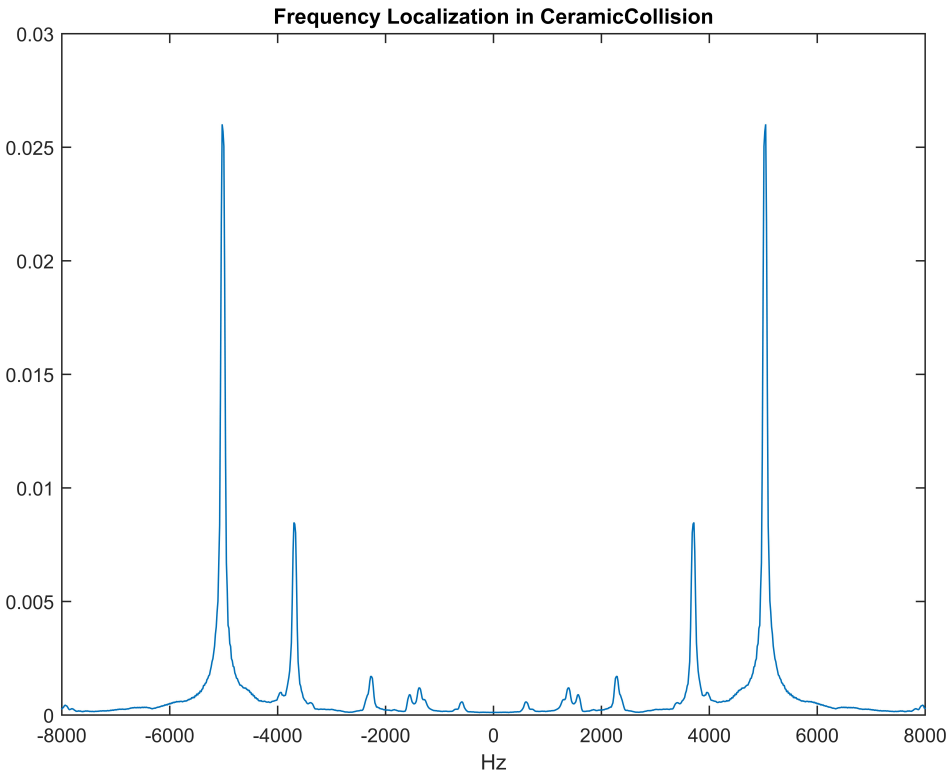

Figure 5.4: CeramicCollision can also be categorized as Frequency Localized Signal

Classifier Training

At this point, the data has been divided into three categories. Recall that the data $D = \{D_U \cup D_L\}$ consists of labeled and unlabeled samples. Hence, we need to assign labels to the unlabeled data. For this, given a set of features $\{F_1, \ldots, F_K\}$, we first select the best features for each category and then learn a Random Forest classifier for each category. Feature set selection is discussed in more detail in Section 5.5.2. It should be noted that the features for this step can be considered as global features as they are sub-framing based features, where each clip is divided into multiple sub-frames and average features over all the sub-frames is used to represent the clip (see Section 2.1). The learning process is iterative wherein at each iteration, the unlabeled samples which

can be predicted with high confidence, are also used to train random forest in the next iteration. The learning process can be described using a block diagram as shown in Figure 5.5.

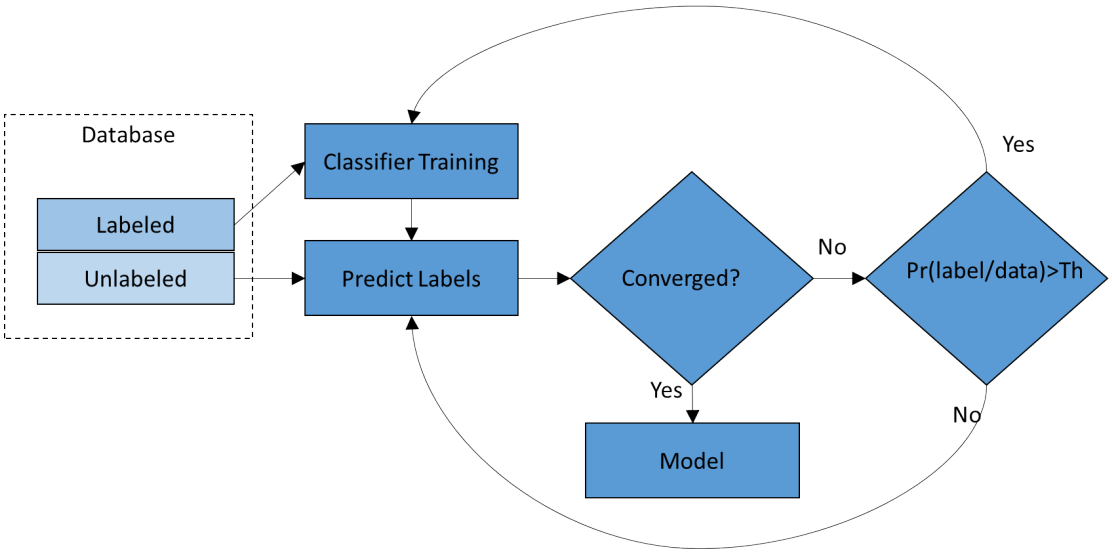

Figure 5.5: Iterative Classifier Training

5.4.2 Stage II

Stage II is the on-line phase of the framework. Given a query, first it is categorized into one of the three categories - Time localized, Frequency Localized and Other, using algorithm discussed in Section 5.4.1. Once the categorization is done, we need to determine the relevant classes for this query. This is described in the following section.

Prospective Class Label Retrieval

Given a test sample, Random Forest Classifier gives a prediction and a set of scores. The scores can be interpreted as probability of the test sample belonging to a particular class. Hence, given classes $C_1, \ldots, C_N$, and a query sample q, we get probabilities $P(C_1/q), \ldots, P(C_N/q)$. Usually, in a classification problem, the class label with highest score is assigned to the test sample. However, in case of retrieval, we need to eventually match and rank the samples in the data-set. If the label prediction is incorrect, there would be no way to obtain the relevant samples from the data set. Hence, instead of assigning just one label, we assign top K labels to the query. This multi-label assignment is done for both the query and the unlabeled samples in the dataset. Hence, this ensures that all the samples belonging to the true class as that of query are deemed as "relevant" for Local Matching and Ranking step.

Local Matching and Ranking

As mentioned before, we used global feature representation for classifier training in Stage I. At this point, we have squeezed out all the information we can from these features. In order to have effective ranking, it is crucial to consider the local variations in a signal. Hence we design a novel bag-of-words feature representation to capture the local information of a signal. In speech recognition, it is ideal to consider a sub-frame size corresponding to $20 - 30msec$ as this time frame is sufficient to capture the structure of phonemes. In environmental sounds, there are no phonemes. However, humans cannot distinguish between sounds of this length. Hence, we can treat features using this frame size. These features capture various signal characteristics over the time period of a sub-frame. Even though the actual signal changes from sub-frame to sub-frame, the distribution of these features can change drastically, or change slowly or not change at all!

Considering this, it would be meaningful to represent audio signal as segments over these features. For this we need to define what an audio segment is. To this end, we propose to use the features of sub-frames as a tool to define and identify audio segments, and represent the audio as a bag-of-words model over these segments. Consider a signal x divided into frames $x_1, \ldots, x_m$ and let $F = \{f_1, \ldots, f_m\}$ be their corresponding features. We can use mean-shift algorithm to first find the modes of the probability density function from which f_i is assumed to sample from. The mean shift procedure is run for each sub-frame f_i:

- Initialize $t = 0$ and $f_i^0 = f_i$
- Compute the mean shift vector:

$$m(f_i^t) = \frac{\sum_{k=1}^{n} g(\frac{f_i^t - f_k}{h_r}) g(\frac{i-k}{h_s}) f_k}{\sum_{k=1}^{n} g(\frac{f_i^t - f_k}{h_s}) g(\frac{i-k}{h_s})} - f_i^t$$

  Note that we are using spatial constraint as well, in order to achieve a better segmentation.

- Move the density estimation window by $m(f_i^t)$, *i.e.* $m(f_i^{t+1}) = m(f_i^t) + f_i^t$
- Repeat until convergence

Figure 5.6 shows MFCC feature for a sample from class AirplaneFlyBy where the top and bottom images are features before and after Mean Shift convergence. It can be seen that the temporal variation in frames has been used to learn segments of audio.

Let s_i be the converged mean for f_i. Then we can cluster all s_i which are close to each other in temporal domain and feature domain using thresholds Th_s and Th_r, respectively. Finally, we obtain segmented representation over cluster centers $z_1, \ldots, z_{m_s}$. Note the number of clusters or unique segments m_s is content dependent

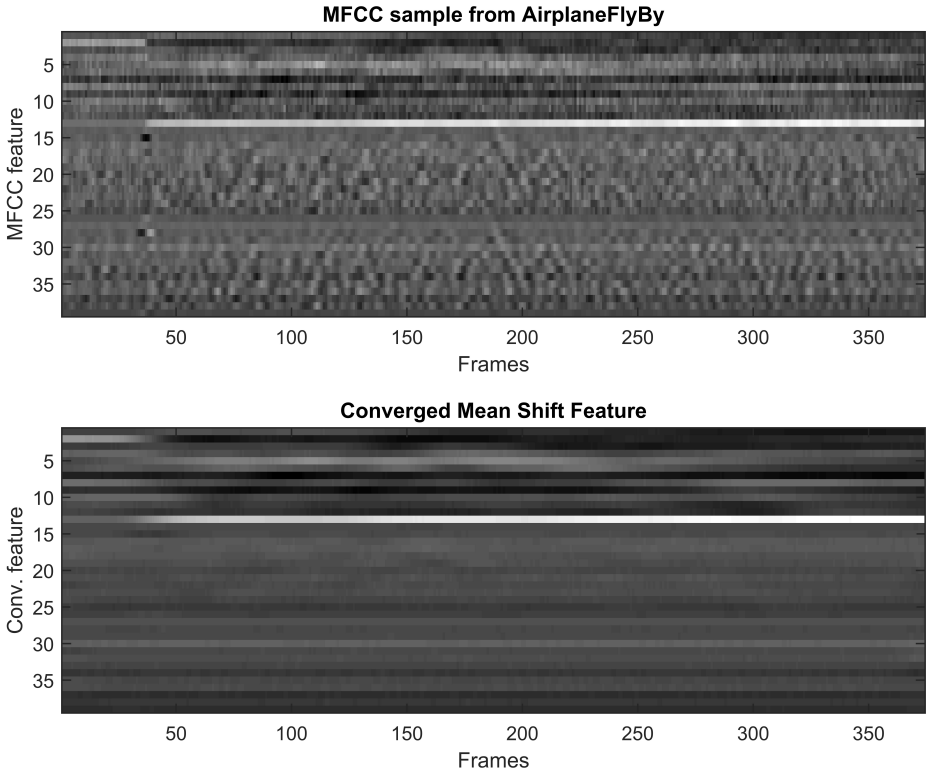

Figure 5.6: Mean Shift Segmentation over MFCC frames for a sample of AirplaneFlyBy

and hence would be different for different signals. This process can be described as follows:

- Starting with a window W_{h_s} of size h_s, and empty set Z
  - Find the mean of all frames in the window: $z_{temp} = \frac{\sum_{k \in W_{h_s}} s_k}{|W_{h_s}|}$
  - If $z_{temp} \notin Z$, $Z = Z \cup z_{temp}$.
- Use k-means algorithm to find $|Z|$ clusters from Z. Let Z_c denote the cluster set.
- Map s_i to Z_c and perform median filtering to remove outlier cluster segments.

Figure 5.7 shows the final clusters that represent the segments of MFCC feature. The bottom image shows the segmentation result of MFCC features.

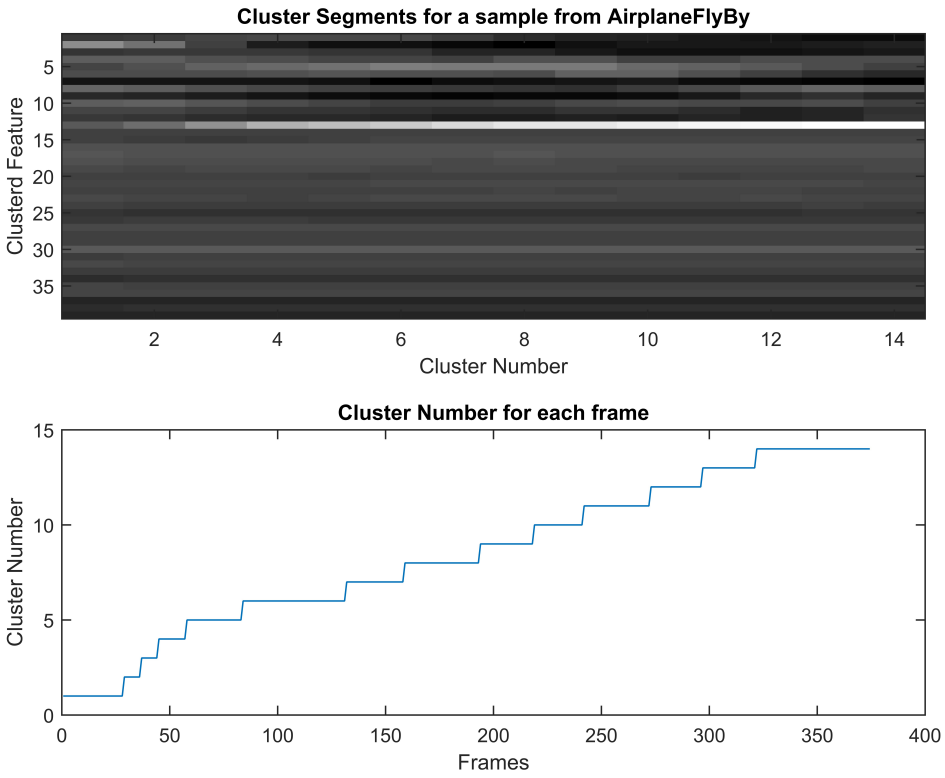

Figure 5.7: Clusters and segmented frames using MFCC feature for a sample of AirplaneFlyBy

With MS-segmentation, we have managed to represent feature F for the signal as a sequence of segments $z_i \in Z_c$. For a traditional bag-of-words representation, we simply need to represent this sequence as a histogram over unique segments. However, all the segments might not have high relevance as fas as comparing two samples of sample class is concerned. For example, Figure 5.8 shows a case where audio sample is represented over 14 clusters; cluster 6 and 14 have almost similar number of segment size, however, the most of the energy in the signal is concentrated in 14 segment. Hence, we propose to represent the signal by the energy of each segment.

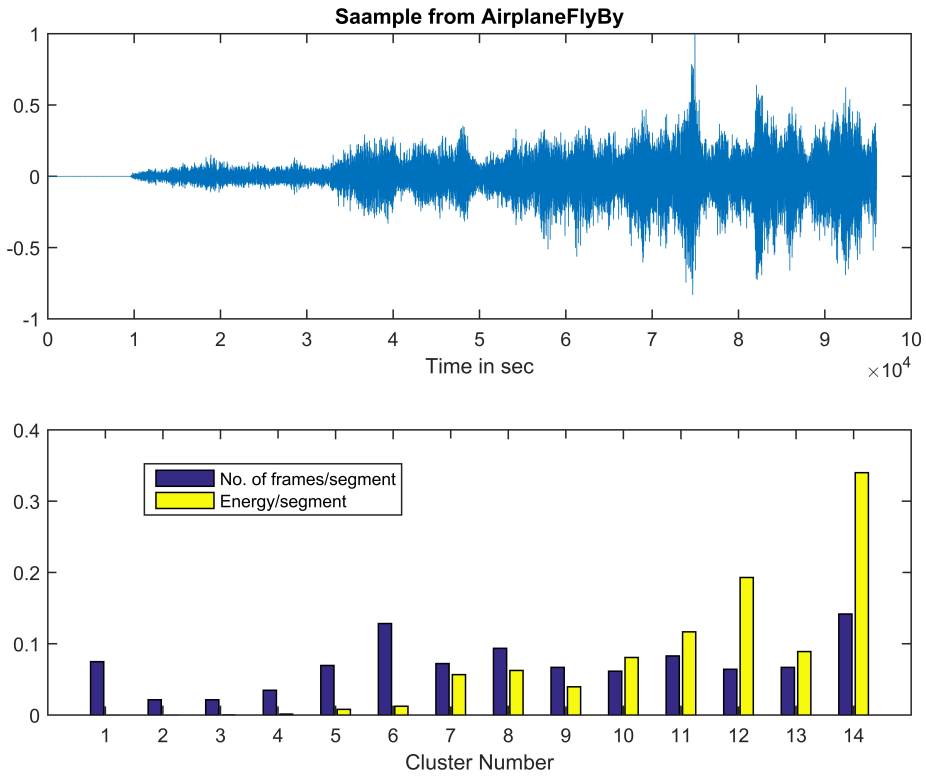

Figure 5.8: Mean Shift Segmented Feature over MFCC

Finally, we need to define a score to rank the documents in the data set against the query. Consider a query q and a document d. Let $Z^q = \{z_1^q, \ldots, z_{m_q}^q\}$ and $Z^d = \{z_1^d, \ldots, z_{m_d}^d\}$ be cluster segments for q and d, respectively. Let $H^q = \{h_1^q, \ldots, h_{m_q}^q\}$ and $H^d = \{h_1^d, \ldots, h_{m_d}^d\}$ be the energy features for q and d, respectively. Then, first a mapping Ψ is done to map the clusters Z^d to clusters Z^q:

$$\Psi(z_i^d, Z^q) = \begin{cases} \text{arg}min_{k\in[1,m_q]}||z_i^d - z_k^q||_2^2, & \text{if } \min_k(||z_i^d - z_k^q||_2^2) < Th \\ \varnothing, & \text{otherwise} \end{cases}$$

Here, $\Psi(z_i^d, Z^q)$ is mapped to $\varnothing$ if z_i^d is not close enough to any of the clusters in Z^q. Once the mapping is done, the energy feature is also mapped to the feature space of Z^q. Given this mapping the score or similarity between q and d can be defined as:

$$Score(q,d) = \sum_{i=1}^{m_q} h_i^q * h^d_{k \ni \Psi(z_k^d, Z^q)=i}.$$

5.5 Experimental Results

5.5.1 Experimental Setup and Data-Set

We investigate the performance of proposed retrieval algorithm on the Environmental Sound Database as described in Section 3.3.1. We choose all the 37 classes from Table 3.2. The set-up is also similar to the one described in Section 3.3.1. For a fair comparison, we used 5-Fold cross validation approach and report all results averaged over the folds.

For Classifier Training step in Stage I, we use following features: NBTF, WPD, NBACF, GWT, MP-Gabor, Modified MP-Gabor, MFCC, NUMAP and Cepstral. For details on the features, please refer to Section 2.2 and Table 3.3. For all features, we use sub-framing based method (see Section 2.1). We take average of feature vectors over sub-frames to obtain a single feature vector to represent one data sample. The feature dimensions per Sample are the final dimensions after any feature selection/dimensionality reduction method has been performed. For Local Ranking and Matching step in Stage II, we use concatenate the feature vectors from all sub-frames to form a matrix and use this for feature extraction. WPD, which consists of a single feature vector representing all the sub-frames, cannot be used for this step.

Table 5.1: Sub-Framing (Section 2.1) for Feature Extraction
(n_f *is number of sub-frames per audio sample)*

	Frame Size ($msec$)	Overlap (%)	Feature Dim. per Sample
NBTF	32	50	20xn_f
WPD	100	80	40x1
NBACF	500	80	192xn_f
GWT	62.5	50	16xn_f
MP-Gabor	32	50	4xn_f
Modified MP-Gabor	32	50	4xn_f
MFCC	32	50	39xn_f
NUMAP	32	50	56xn_f
CEPSTRAL	32	50	20xn_f

For evaluation, we use the Bull's eye score, *i.e.* retrieval accuracy when the number of documents retrieve is twice that of the number of documents belonging to the true class in the data-base. We also use precision and recall curves to analyze the results.

5.5.2 Results and Discussion

Table 5.2 shows the bull's eye score for proposed method with CEPSTRAL, NBTF, MFCC and NBACF features. It also compares their performance with the rank fusion approach. Among individual features, NBTF gives the best perormance consistently for all cases with varying amount of unlabeled data. This shows that the our proposed NBTF features, are most efficient in representing the dyamic changes in signal over sub-frames. Performance of MFCC and Cepstral features is similar. NBACF scores trail those of others. Recall that NBTF and NBACF features are non-stationary, while MFCC and Cepstral are stationary features. The dynamic feature extraction method proposed in Section 5.4.2 and non-stationary feature representation for sub-frames give the best performance owing to their ability to represent non-stationarity at both frame

Table 5.2: Bull's Eye Scores

Method	Percentage of Unlabeled Data				
	10	30	50	70	90
CEPSTRAL	0.7485	0.7444	0.7399	0.7284	0.6927
NBTF	0.7779	0.7744	0.7684	0.7558	0.7202
MFCC	0.7523	0.7506	0.7439	0.7312	0.6955
NBACF	0.7098	0.7100	0.7015	0.6883	0.6484
Rank Fusion	0.8040	0.8003	0.7948	0.7829	0.7488

and sub-frame levels. Finally, ranking fusion improves the results further and gives best performance. Even with 90% of unlabeled data, we obtain a bull's eye score of 0.7488. Performance and recall curves for 10%, 50% and 90% unlabeled data has been shown in Figures 5.9, 5.10 and 5.11, respectively. Clearly, rank fusion approach improves the performance for any number of documents retrieved.

In Figure 5.12, we show the Bull's Eye Scores for each category over different features. For Time Localized signals, NBTF gives the best performance. Recall that the time localized signals are ones with concentrated energy in a small time frame, and hence have very few number of sub-frames per sample. Hence, it is intuitive that the non-stationary NBTF feature would be most discriminative as, with small number of sub-frames, non-stationarity characteristics of sub-frame are more important than the temporal dynamic over different sub-frames. For Frequency localized signals, Cepstral features give the best performance. Again, this is not surprising given that this category is defined by its frequency localized characteristics and cepstral features capture this information adeptly. For Other category, NBTF gives the best performance. It can also be seen that the overall performance for Other category is lowest as it contains most diverse classes. Over all, it is clear that Broad Categorization in Stage I is very helpful to improve the overall performance of the system.

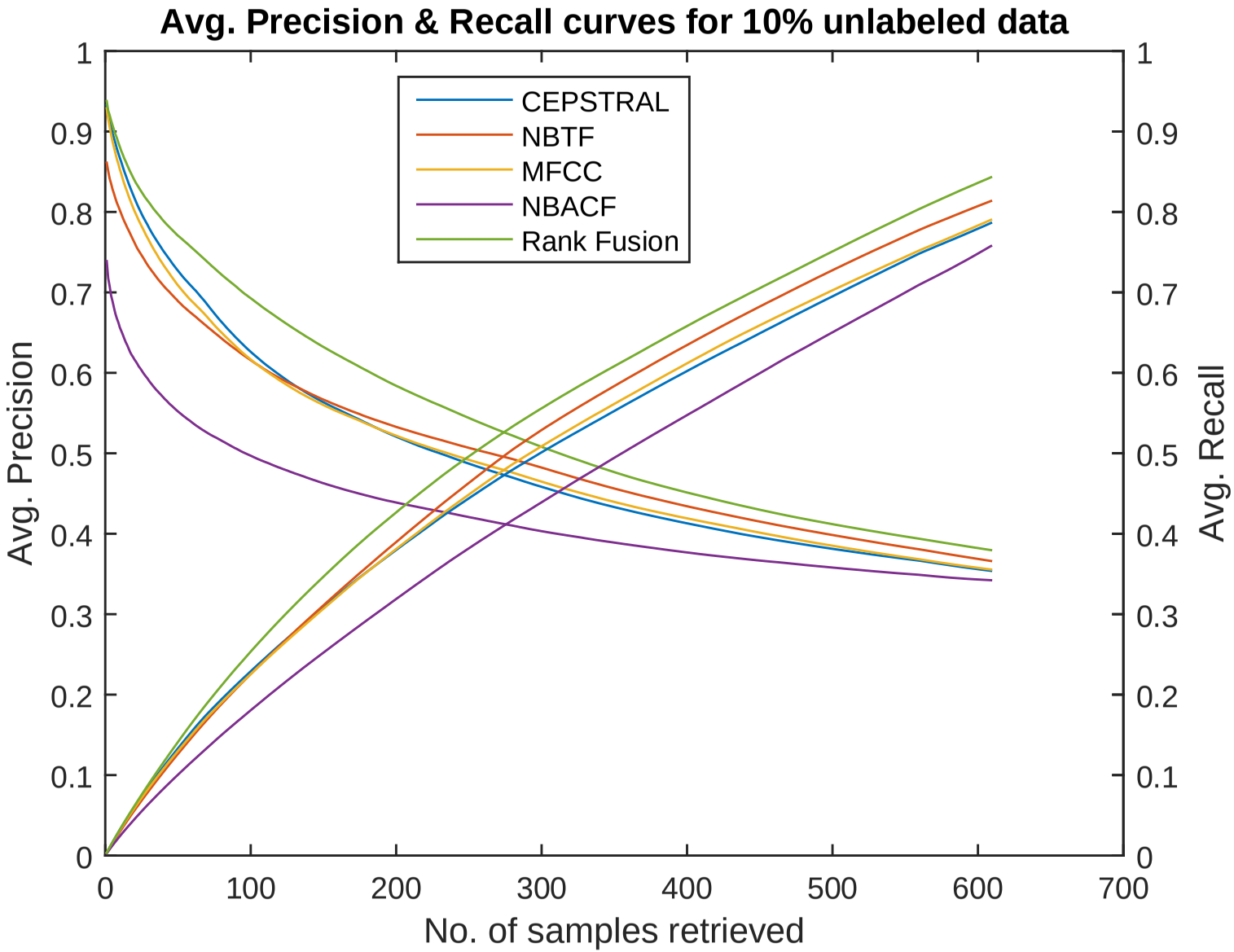

Figure 5.9: Avg. Precision and Recall curves for 10% unlabeled data

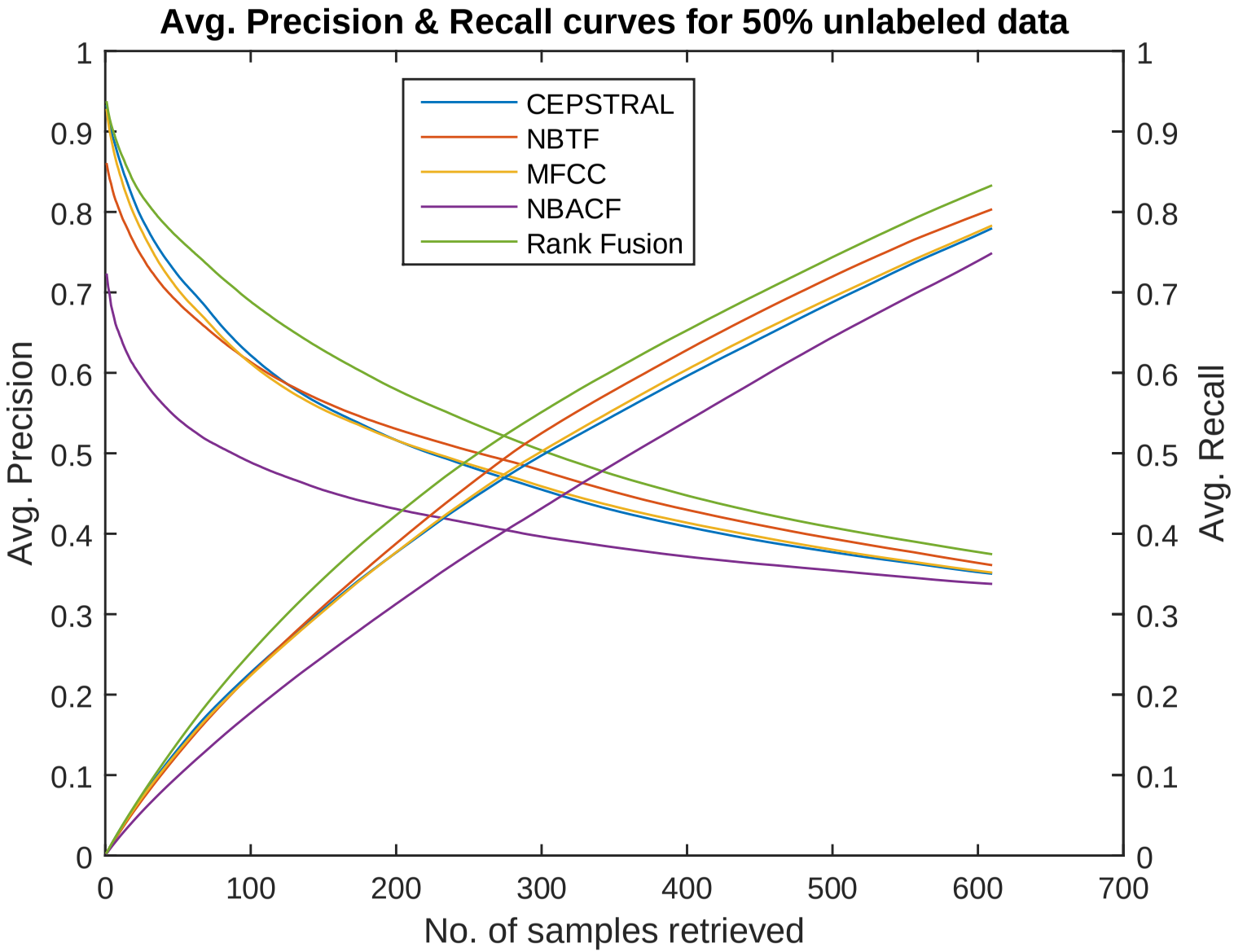

Figure 5.10: Avg. Precision and Recall curves for 50% unlabeled data

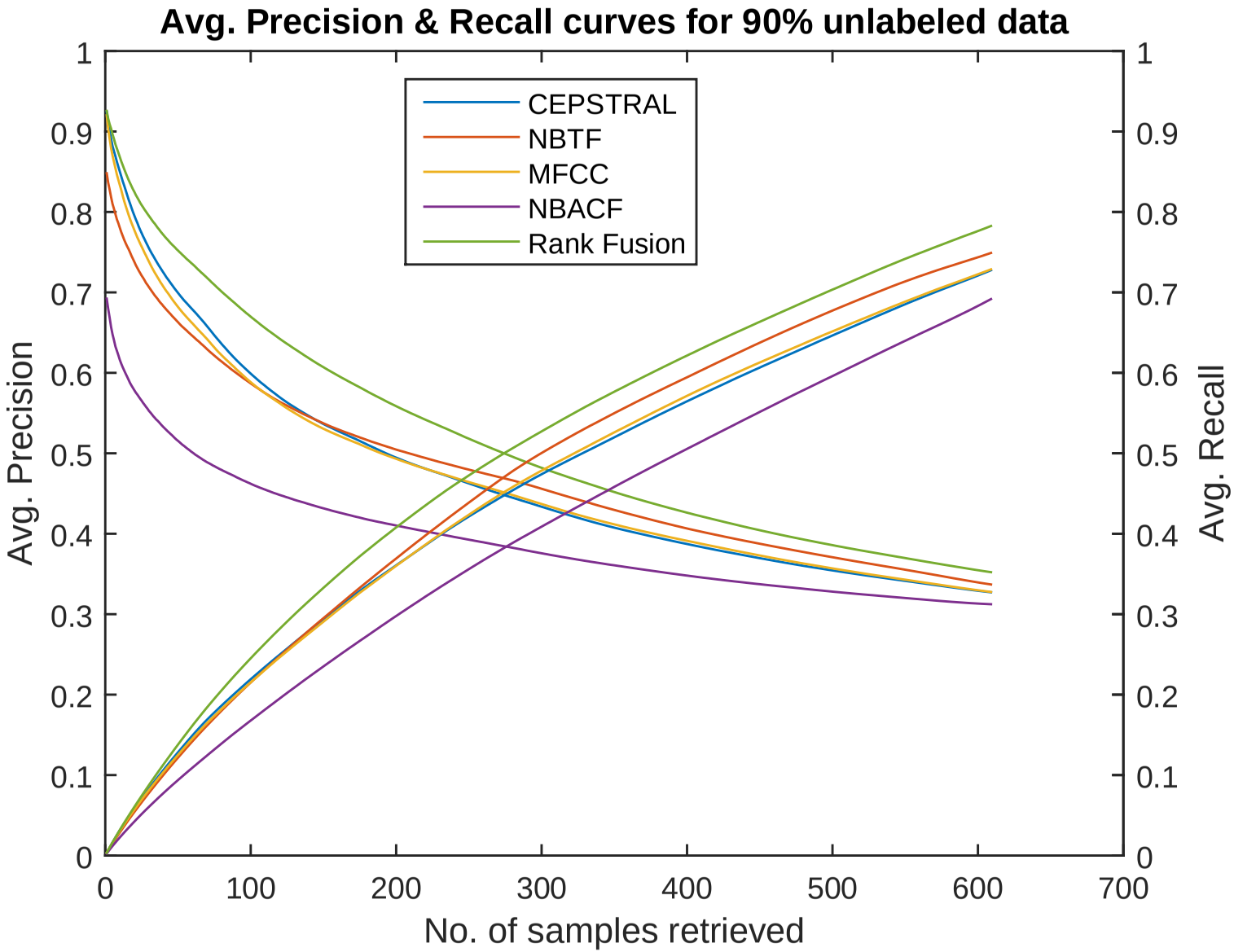

Figure 5.11: Avg. Precision and Recall curves for 90% unlabeled data

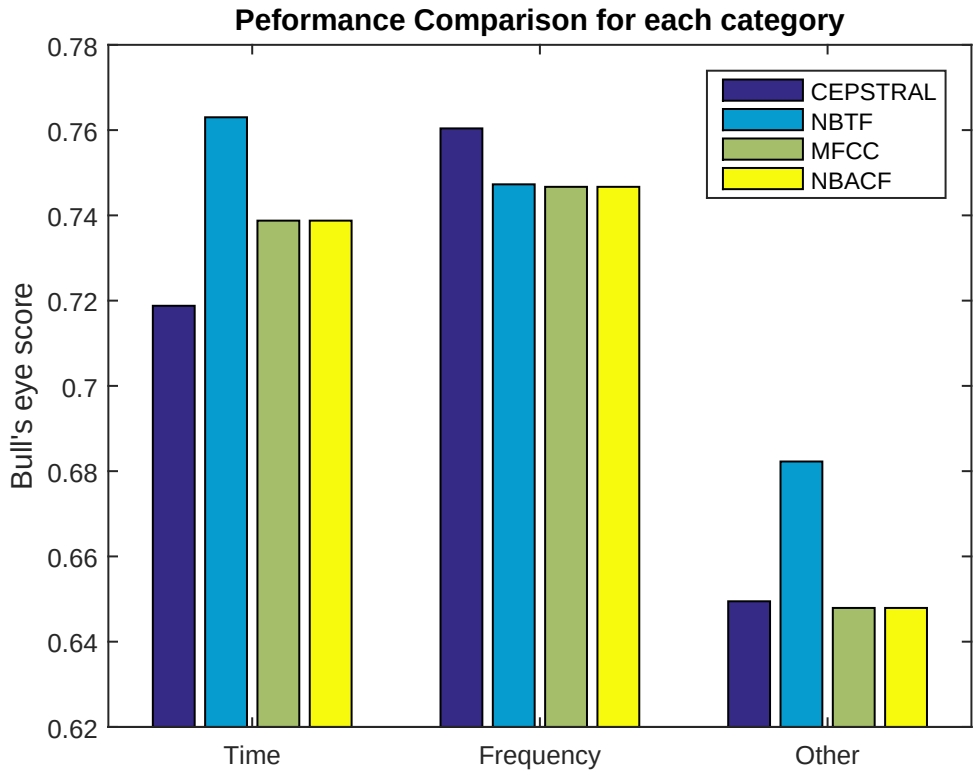

Figure 5.12: Bull's Eye Score comparison for different categories and features for the case of 90% unlabeled data

For comparison, in Figure 5.13 we show the precision and recall curves for traditional method which assumes no label information in the database. The best bull's eye score we get for traditional methods is 0.4085. Here, the stationary features outperform the non-stationary features. This is non-surprising given the fact that the non-stationarity captured in sub-frames is lost due to averaging over sub frames despite the dynamic variations over sub-frames. Overall, the performance is poor and much less than that of the proposed method.

In Figures 5.14, 5.14 and 5.14, we analyze the classifier training and features for Time Localized, Frequency Localized and Other categories, respectively. For Time

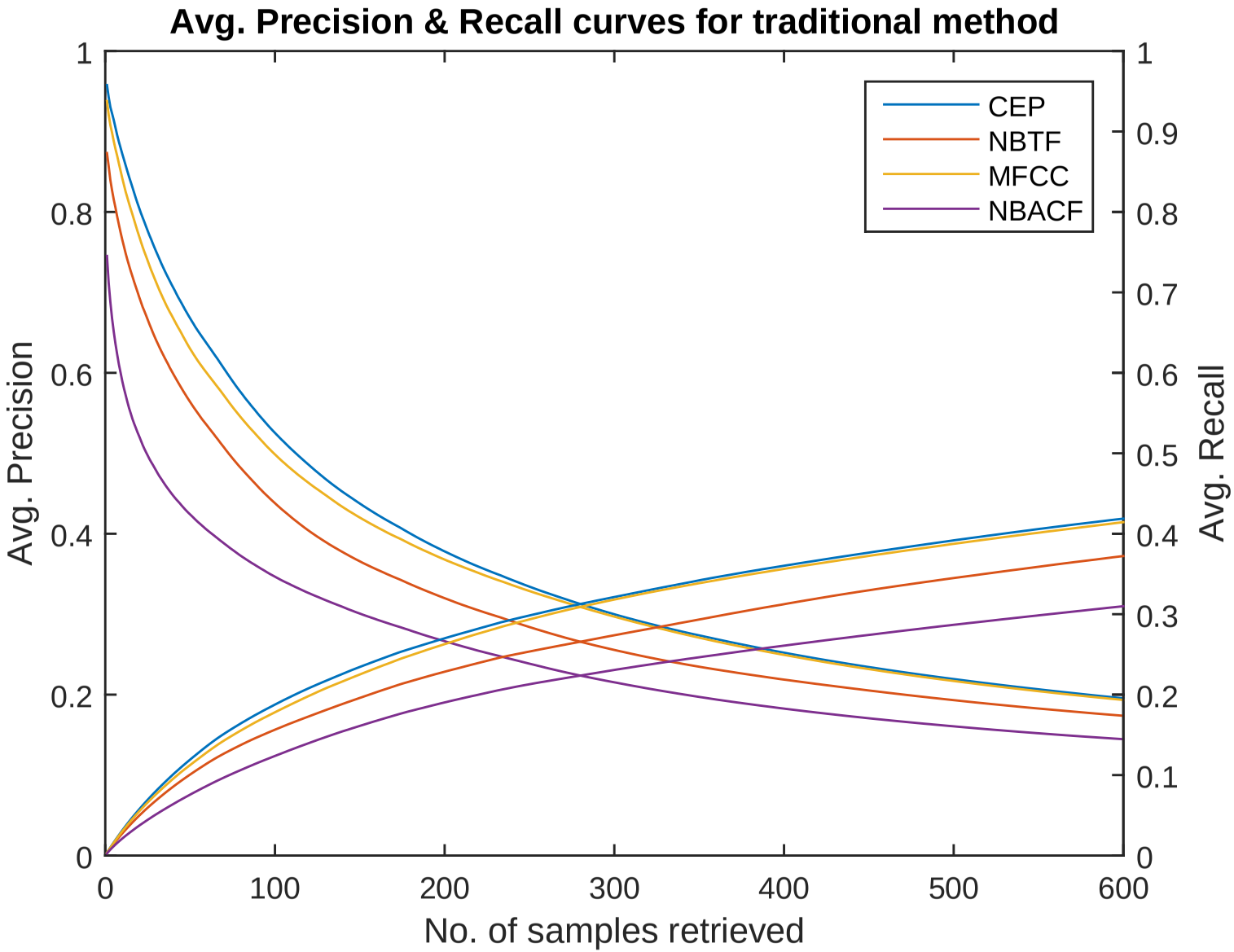

Figure 5.13: Avg. Precision and Recall curves for traditional methods

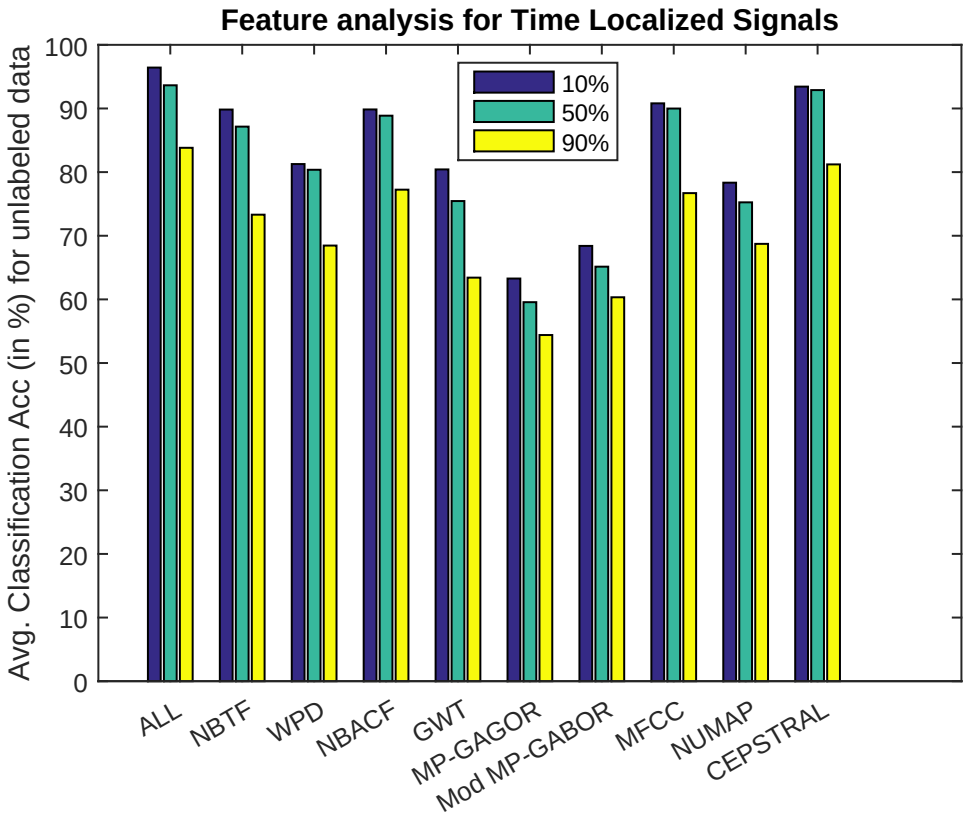

Figure 5.14: Feature analysis for Classifier Training in Stage I for Time Localized signals

Localized signals and Other category signals, cepstral features and all features concatenated together give comparable performance. Given the complexity of Random Forest classification, we choose to use only Cepstral Features for Time Localized signals in Stage I. For Frequency Localized signals, we use all the features concatenated together as this gives the best classification results. It should be noted that the features selected for each case are chosen by keeing in mind the robustness of classifier for large amount of unseen data.

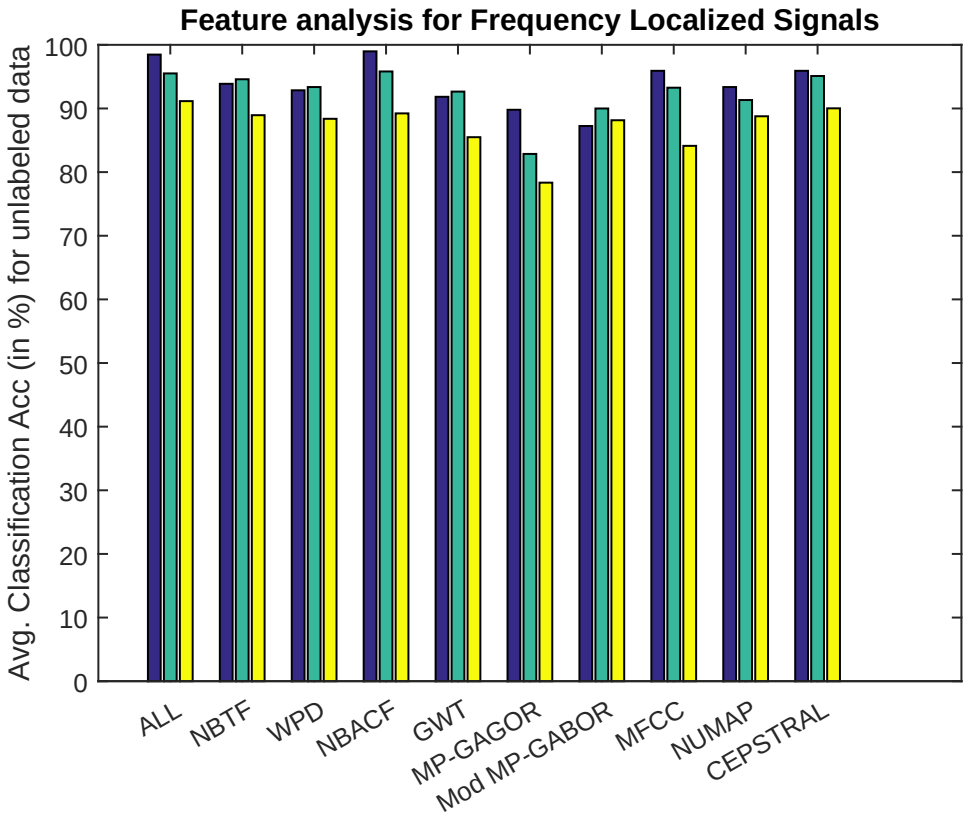

Figure 5.15: Feature analysis for Classifier Training in Stage I for Frequency Localized signals

5.6 Conclusion

In this chapter, we proposed a novel two stage framework for Environmental Sound retrieval. We also proposed a feature and scoring method for matching and ranking two environmental sounds. Finally, we use rank fusion approach to improve the results. The best Bull's eye score achieved for 90% unlabeled data is 0.7488 while that for 10% unlabeled data is 0.8040. We also analyzed the performance of various stationary and non-stationary features with respect to their capability of modeling dynamic changes in audio signal not only over sub-frames, but also over multiple sub-frames for an environmental sound signal.

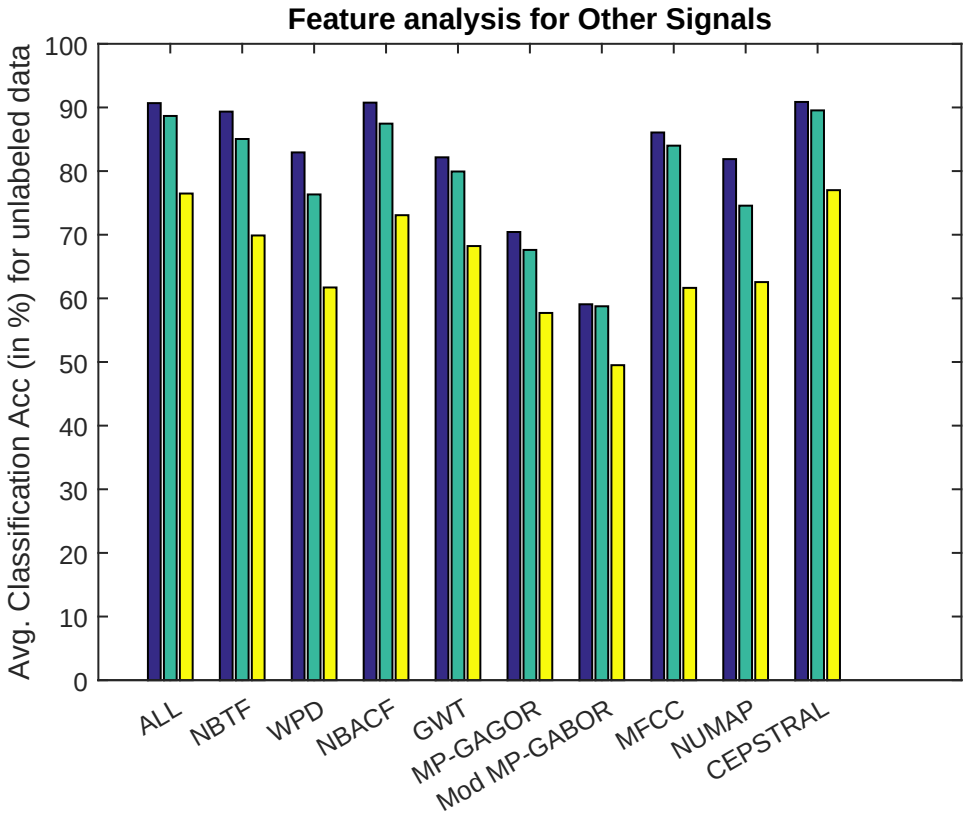

Figure 5.16: Feature analysis for Classifier Training in Stage I for Other category signals

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this dissertation, we extensively studied the content based modeling of Environmental Sounds for the purposes of recognition and retrieval.

In Chapter 3, we first studied the non-stationary characteristics of environmental sounds. We compared the performance of several state of the art approaches on a common platform using our own ESR database. We provide a detailed critique on the performance of these features. We also proposed a new set of features, Narrow Band Time Frequency features which use dual-representation of narrow-band filtered signal to capture its non-stationarity. With the use of MFCC filter-bank, we ensure meaningful band separation. Sparse representation of these filtered signals over Gabor Dictionary ensures that maximum information about signal's statistical characteristics over both time and frequency domain is represented in the feature set. We compared the representation capability of NBTF features with most-state of the art approaches and showed its superior performance. We were able to achieve a significant improvement over the performance of the best feature in benchmark models, i.e. MFCC. Finally, we showed that the NBTF features and MCC features are complementary to each other and hence can be used together to improve the results further.

To further improve the results of Environmental Sound Recognition, in Chapter 4 we proposed a novel Multi-Classifier System - Para-Boost Multi Classifier System. It combines the ideas of Stacked Generalization and Random Subspace methods to exploit

the existing diversity in vertically decomposed experts consisting of different feature sets. In total, we proposed 4 different variations of PB (PB1-PB4) based on how features we coded for meta-classifier. We reported that one of the coding schemes, PB1, which uses all the score outputs from individual experts, outperforms other coding schemes. We then proposed two more variations - Horizontally decomposed PB and Grouping based PB. HBP not only leverages the advantages of vertical decomposition, but also used Horizontal decomposition to improve the performance. GPB, on the other hand, exploits inherent structure of dataset to partition it into smaller sets and model PB for each individually. Both HPB and GBP provide significant improvement over individual experts and also perform better than benchmark methods - DCSLA and BKS.

In Chapter 5, we tackle the problem of search and retrieval of environmental sounds. We proposed a novel Two Stage Content-Based Environmental Sound Retrieval System for partially labeled database. The proposed framework leverages signal characteristics in time and frequency domain to categorize the database into three different categories - Time Localized, Frequency Localized and Other Signals. We used category dependent classifiers to predict class labels for unlabeled data. A variant of bag-of-words representation was proposed to model a query and documents in the database. For this, we used Mean Shift segmentation to first segment an audio signal into meaningful self-contained segments. We then represent the audio signal as a feature with energy distribution over these segments. Finally, we proposed a scoring mechanism which first mapped documents on database to the query, and then scored the overlapping segments between the two as a measure of similarity. We compared the performance of this retrieval system with traditional Euclidean metric based brute force approach and showed that the proposed framework almost doubles the bull's eye score. We proposed to improve the results further using rank fusion.

6.2 Future Research Directions

In this work, we considered environmental sounds that were either isolated sound events or ambiance sounds. In either cases, the sound is considered as an entity in itself and the database consisted of only one entity at a time. For future work, we would like to extend the work of Environmental Sound Recognition to heterogeneous database where data can consist of more than one overlapping sound classes. This is analogues to an open problem in speech recognition task when more than one speakers are speaking at the same time. ESR in itself is a challenging problem due to non-stationary characteristics of environmental sounds. Overlapping these sounds in time domain makes the problem much more harder. However, for ESR to be useful for broad set of applications, it needs to be capable of handling heterogeneous sounds. Hence, this would be an interesting research direction.

There is large amount of environmental sound data on the Internet with contextual information. We would like to leverage this information, along with content based retrieval system, to form a powerful retrieval system. This would entail not only information fusion, but also linking the contextual and content-based information so that contextual information can be predicted from content it self, rather than relying on subjective and non-expert tagging done by users.

We hope that this dissertation would eventually enable to build better applications involving environmental sounds.

Bibliography

[1] BAILLIE, M., AND JOSE, J. M. Audio-based event detection for sports video. In *Image and Video Retrieval*. Springer, 2003, pp. 300–309.

[2] BARDELI, R. Similarity search in animal sound databases. *Multimedia, IEEE Transactions on 11*, 1 (2009), 68–76.

[3] BARDELI, R., WOLFF, D., KURTH, F., KOCH, M., TAUCHERT, K.-H., AND FROMMOLT, K.-H. Detecting bird sounds in a complex acoustic environment and application to bioacoustic monitoring. *Pattern Recognition Letters 31*, 12 (2010), 1524–1534.

[4] BRIGGS, F., LAKSHMINARAYANAN, B., NEAL, L., FERN, X. Z., RAICH, R., HADLEY, S. J., HADLEY, A. S., AND BETTS, M. G. Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach. *The Journal of the Acoustical Society of America 131*, 6 (2012), 4640–4650.

[5] BROWN, G., WYATT, J., HARRIS, R., AND YAO, X. Diversity creation methods: a survey and categorisation. *Information Fusion 6*, 1 (2005), 5–20.

[6] CHANG, C.-C., AND LIN, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology 2* (2011), 27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[7] CHEN, J., KAM, A. H., ZHANG, J., LIU, N., AND SHUE, L. Bathroom activity monitoring based on sound. In *Pervasive Computing*. Springer, 2005, pp. 47–61.

[8] CHU, S., NARAYANAN, S., AND KUO, C.-C. J. Environmental sound recognition with time–frequency audio features. 1142–1158.

[9] CHU, S., NARAYANAN, S., KUO, C.-C. J., AND MATARIC, M. J. Where am I? Scene recognition for mobile robots using audio features. In *Multimedia and Expo, 2006 IEEE International Conference on* (2006), IEEE, pp. 885–888.

[10] CHUI, C. K. *An Introduction to Wavelets*, vol. 1. Academic press, 1992.

[11] COWLING, M., AND SITTE, R. Comparison of techniques for environmental sound recognition. *Pattern Recognition Letters 24*, 15 (2003), 2895–2907.

[12] CRISTANI, M., BICEGO, M., AND MURINO, V. Audio-visual event recognition in surveillance video sequences. 257–267.

[13] DENG, J. D., SIMMERMACHER, C., AND CRANEFIELD, S. A study on feature analysis for musical instrument classification. 429–438.

[14] DIETTERICH, T. G. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation 10*, 7 (1998), 1895–1923.

[15] DUAN, S., ZHANG, J., ROE, P., AND TOWSEY, M. A survey of tagging techniques for music, speech and environmental sound. *Artificial Intelligence Review* (2012), 1–25.

[16] FOOTE, J. T. Content-based retrieval of music and audio. In *Voice, Video, and Data Communications* (1997), International Society for Optics and Photonics, pp. 138–147.

[17] GHORAANI, B., AND KRISHNAN, S. Time–frequency matrix feature extraction and classification of environmental audio signals. 2197–2209.

[18] GHORAANI, B., AND KRISHNAN, S. Discriminant non-stationary signal features' clustering using hard and fuzzy cluster labeling. *EURASIP Journal on Advances in Signal Processing 2012*, 1 (2012), 250.

[19] GHOSAL, A., CHAKRABORTY, R., DHARA, B. C., AND SAHA, S. K. Song / instrumental classification using spectrogram based contextual features. In *Proceedings of the CUBE International Information Technology Conference* (2012), ACM, pp. 21–25.

[20] GROSSMANN, A., AND MORLET, J. Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM journal on mathematical analysis 15*, 4 (1984), 723–736.

[21] GUNASEKARAN, S., AND REVATHY, K. Content-based classification and retrieval of wild animal sounds using feature selection algorithm. In *machine learning and computing (ICMLC), 2010 second international conference on* (2010), IEEE, pp. 272–275.

[22] HAN, B.-J., AND HWANG, E. Environmental sound classification based on feature collaboration. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on* (2009), IEEE, pp. 542–545.

[23] KARBASI, M., AHADI, S., AND BAHMANIAN, M. Environmental sound classification using spectral dynamic features. In *Information, Communications and Signal Processing (ICICS) 2011 8th International Conference on* (2011), IEEE, pp. 1–5.

[24] KHUNARSAL, P., LURSINSAP, C., AND RAICHAROEN, T. Very short time environmental sound classification based on spectrogram pattern matching. 2013.

[25] KIM, S. *Contextual modeling of audio signals toward information retrieval*. PhD thesis, University of Southern California, 2010.

[26] KIM, S., GEORGIOU, P. G., NARAYANAN, S., AND SUNDARAM, S. Supervised acoustic topic model for unstructured audio information retrieval. In *Proceedings of Asia Pacific Signal and Information Processing Association (APSIPA) Annual Summit and Conference* (2010), vol. 3, Citeseer, p. 3.

[27] LAZARIDIS, M., AXENOPOULOS, A., RAFAILIDIS, D., AND DARAS, P. Multimedia search and retrieval using multimodal annotation propagation and indexing techniques. *Signal Processing: Image Communication 28*, 4 (2013), 351–367.

[28] LEE, C.-H., HAN, C.-C., AND CHUANG, C.-C. Automatic classification of bird species from their sounds using two-dimensional cepstral coefficients. *Audio, Speech, and Language Processing, IEEE Transactions on 16*, 8 (2008), 1541–1550.

[29] LIU, H., MOTODA, H., SETIONO, R., AND ZHAO, Z. Feature selection: An ever evolving frontier in data mining. In *Proc. The Fourth Workshop on Feature Selection in Data Mining* (2010), vol. 4, pp. 4–13.

[30] MECHTLEY, B. M. *Techniques for Soundscape Retrieval and Synthesis*. PhD thesis, Arizona State University, 2013.

[31] MITROVIC, D., ZEPPELZAUER, M., AND BREITENEDER, C. Discrimination and retrieval of animal sounds. In *Multi-Media Modelling Conference Proceedings, 2006 12th International* (2006), IEEE, pp. 5–pp.

[32] MITROVIĆ, D., ZEPPELZAUER, M., AND BREITENEDER, C. Features for content-based audio retrieval. *Advances in computers 78* (2010), 71–150.

[33] MUHAMMAD, G., ALOTAIBI, Y. A., ALSULAIMAN, M., AND HUDA, M. N. Environment recognition using selected MPEG-7 audio features and Mel-Frequency Cepstral Coefficients. In *Digital Telecommunications (ICDT), 2010 Fifth International Conference on* (2010), IEEE, pp. 11–16.

[34] NAKAMURA, S., HIYANE, K., ASANO, F., NISHIURA, T., AND YAMADA, T. Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition. In *LREC* (2000).

[35] PELTONEN, V., TUOMI, J., KLAPURI, A., HUOPANIEMI, J., AND SORSA, T. Computational auditory scene recognition. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on* (2002), vol. 2, IEEE, pp. II–1941.

[36] PICKENS, J. A survey of feature selection techniques for music information retrieval, 2001.

[37] POTAMITIS, I., AND GANCHEV, T. Generalized recognition of sound events: Approaches and Applications. In *Multimedia Services in Intelligent Environments*. Springer, 2008, pp. 41–79.

[38] RABINER, L. R. A tutorial on hidden markov models and selected applications in speech recognition. 257–286.

[39] RAFAILIDIS, D., MANOLOPOULOU, S., AND DARAS, P. A unified framework for multimodal retrieval. *Pattern Recognition 46*, 12 (2013), 3358–3370.

[40] SADLIER, D., O'CONNOR, N. E., ET AL. Event detection in field sports video using audio-visual features and a support vector machine. *Circuits and Systems for Video Technology, IEEE Transactions on 15*, 10 (2005), 1225–1233.

[41] SCARINGELLA, N., ZOIA, G., AND MLYNEK, D. Automatic genre classification of music content: a survey. 133–141.

[42] SITTE, R., AND WILLETS, L. Non-speech environmental sound identification for surveillance using self-organizing-maps. In *Proceedings of the Fourth conference on IASTED International Conference: Signal Processing, Pattern Recognition, and Applications* (Anaheim, CA, USA, 2007), SPPR'07, ACTA Press, pp. 281–286.

[43] SIVASANKARAN, S., AND PRABHU, K. Robust features for environmental sound classification. In *Electronics, Computing and Communication Technologies (CONECCT), 2013 IEEE International Conference on* (2013), pp. 1–6.

[44] SOULI, S., AND LACHIRI, Z. Environmental sounds classification based on visual features. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Springer, 2011, pp. 459–466.

[45] SOULI, S., AND LACHIRI, Z. Environmental sounds spectrogram classification using log-gabor filters and multiclass support vector machines. *arXiv preprint arXiv:1209.5756* (2012).

[46] SU, F., YANG, L., LU, T., AND WANG, G. Environmental sound classification for scene recognition using local discriminant bases and HMM. In *Proceedings of the 19th ACM international conference on Multimedia* (2011), ACM, pp. 1389–1392.

[47] TSAU, E., KIM, S.-H., AND KUO, C.-C. J. Environmental sound recognition with CELP-based features. In *Signals, Circuits and Systems (ISSCS), 2011 10th International Symposium on* (2011), IEEE, pp. 1–4.

[48] UMAPATHY, K., KRISHNAN, S., AND RAO, R. K. Audio signal feature extraction and classification using local discriminant bases. 1236–1246.

[49] VACHER, M., PORTET, F., FLEURY, A., AND NOURY, N. Challenges in the processing of audio channels for ambient assisted living. In *e-Health Networking Applications and Services (Healthcom), 2010 12th IEEE International Conference on* (2010), IEEE, pp. 330–337.

[50] VALERO, X., AND ALÍAS, F. Classification of audio scenes using narrow-band autocorrelation features. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European* (2012), IEEE.

[51] VALERO, X., AND ALÍAS, F. Gammatone wavelet features for sound classification in surveillance applications. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European* (2012), IEEE, pp. 1658–1662.

[52] VAN DER MAATEN, L., POSTMA, E., AND VAN DEN HERIK, H. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research 10* (2009), 1–41.

[53] VIRTANEN, T., AND HELÉN, M. Probabilistic model based similarity measures for audio query-by-example. In *Applications of Signal Processing to Audio and Acoustics, 2007 IEEE Workshop on* (2007), IEEE, pp. 82–85.

[54] WANG, J.-C., LEE, H.-P., WANG, J.-F., AND LIN, C.-B. Robust environmental sound recognition for home automation. 25–31.

[55] WANG, J.-C., LIN, C.-H., CHEN, B.-W., AND TSAI, M.-K. Gabor-based nonuniform scale-frequency map for environmental sound classification in home automation. *Automation Science and Engineering, IEEE Transactions on 11*, 2 (April 2014), 607–613.

[56] WANG, J.-C., WANG, J.-F., HE, K. W., AND HSU, C.-S. Environmental sound classification using hybrid SVM/KNN classifier and MPEG-7 audio low-level descriptor. In *Neural Networks, 2006. IJCNN'06. International Joint Conference on* (2006), IEEE, pp. 1731–1735.

[57] WENINGER, F., AND SCHULLER, B. Audio recognition in the wild: static and dynamic classification on a real-world database of animal vocalizations. In *acoustics, speech and signal processing (ICASSP), 2011 IEEE international conference on* (2011), IEEE, pp. 337–340.

[58] WICHERN, G., XUE, J., THORNBURG, H., MECHTLEY, B., AND SPANIAS, A. Segmentation, indexing, and retrieval for environmental and natural sounds. *Audio, Speech, and Language Processing, IEEE Transactions on 18*, 3 (2010), 688–707.

[59] WOODS, K., BOWYER, K., AND KEGELMEYER JR, W. P. Combination of multiple classifiers using local accuracy estimates. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on* (1996), IEEE, pp. 391–396.

[60] YAMAKAWA, N., KITAHARA, T., TAKAHASHI, T., KOMATANI, K., OGATA, T., AND OKUNO, H. G. Effects of modelling within-and between-frame temporal variations in power spectra on non-verbal sound recognition. In *Proc. 2010 International Conference on Spoken Language Processing, Makuhari* (2010), Citeseer, pp. 2342–2345.

[61] YAMAKAWA, N., TAKAHASHI, T., KITAHARA, T., OGATA, T., AND OKUNO, H. G. Environmental sound recognition for robot audition using Matching-Pursuit. In *Modern Approaches in Applied Intelligence*. Springer, 2011, pp. 1–10.

[62] YU, G., AND SLOTINE, J.-J. Fast wavelet-based visual classification. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* (2008), pp. 1–5.

[63] YU, G., AND SLOTINE, J.-J. Audio classification from time-frequency texture. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on* (2009), IEEE, pp. 1677–1680.

[64] ZHANG, T., AND KUO, C. J. Hierarchical classification of audio data for archiving and retrieving. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on* (1999), vol. 6, IEEE, pp. 3001–3004.