School of Engineering

Audio Surveillance in Unstructured Environments

Roneel Vikash Sharan

2015

A thesis submitted to Auckland University of Technology in fulfillment of the requirements for the degree of Doctor of Philosophy (PhD)

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

……

Roneel Vikash Sharan

Abstract

This research examines an audio surveillance application, one of the many applications of sound event recognition (SER), and aims to improve the sound recognition rate in the presence of environmental noise using time-frequency image analysis of the sound signal and deep learning methods. The sound database contains ten sound classes, each sound class having multiple subclasses with interclass similarity and intraclass diversity. Three different noise environments are added to the sound signals and the proposed and baseline methods are tested under clean conditions and at four different signal-to-noise ratios (SNRs) in the range of 0–20dB.

A number of baseline features are considered in this work which are mel-frequency cepstral coefficients (MFCCs), gammatone cepstral coefficients (GTCCs), and the spectrogram image feature (SIF), where the sound signal spectrogram images are divided in blocks, central moments are computed in each block and concatenated to form the final feature vector. Next, several methods are proposed to improve the classification performance in the presence of noise.

Firstly, a variation of the SIF with reduced feature dimensions is proposed, referred as the reduced spectrogram image feature (RSIF). The RSIF utilizes the mean and standard deviation of the central moment values along the rows and columns of the blocks resulting in a 2.25 times lower feature dimension than the SIF. Despite the reduction in feature dimension, the RSIF was seen to outperform the SIF in classification performance due to its higher immunity to inconsistencies in sound signal segmentation.

Secondly, a feature based on the image texture analysis technique of gray-level co-occurrence matrix (GLCM) is proposed, which captures the spatial relationship of pixels in an image. The GLCM texture analysis technique is applied in subbands to the spectrogram image and the matrix values from each subband are concatenated to form the final feature vector which is referred as the spectrogram image texture

feature (SITF). The SITF was seen to be significantly more noise robust than all the baseline features and the RSIF, but with a higher feature dimension.

Thirdly, the time-frequency image representation called cochleagram is proposed over the conventional spectrogram images. The cochleagram image is a variation of the spectrogram image utilizing a gammatone filter, as used for GTCCs. The gammatone filter offers more frequency components in the lower frequency range with narrow bandwidth and less frequency components in the higher frequency range with wider bandwidth which better reveals the spectral information for the sound signals considered in this work. With cochleagram feature extraction, the spectrogram features SIF, RSIF, and SITF are referred as CIF, RCIF, and CITF, respectively. The use of cochleagram feature extraction was seen to improve the classification performance under all noise conditions with the most improved results at low SNRs.

Fourthly, feature vector combination has been seen to improve the classification performance in a number of literature and this work proposes a combination of linear GTCCs and cochleagram image features. This feature combination was seen to improve the classification performance of CIF, RCIF, and CITF and, once again, the most improved results were at low SNRs.

Finally, while support vector machines (SVMs) seem to be the preferred classifier in most SER applications, deep neural networks (DNNs) are proposed in this work. SVMs are used as a baseline classifier but in each case the results are compared with DNNs. SVM being a binary classifier, four common multiclass classification methods, one-against-all (OAA), one-against-one (OAO), decision directed acyclic graph (DDAG), and adaptive directed acyclic graph (ADAG), are considered. The classification performance of all the classification methods is compared with individual and combined features and the training and evaluation times are also compared. For the multiclass SVM classification methods, the OAA method was generally seen to be the most noise robust and gave a better overall classification performance. However, the noise robustness of the DNN classifier was determined to be the best together with the best overall classification performance with both individual and combined features. DNNs also offered the fastest evaluation time but the training time was determined to be the slowest.

Acknowledgements

I would like to thank the AUT scholarships committee for awarding me the Queen Elizabeth II Diamond Jubilee Doctoral Scholarship through which my PhD study was made possible. A big thanks also to my supervisors, parents, brother, and all other family and friends for their support through this journey.

List of Publications

Journal Papers

1. R. V. Sharan and T. J. Moir, "An overview of applications and advancements in automatic sound recognition," *Neurocomputing*, vol. 200, pp. 22–34, 2016.
2. R. V. Sharan and T. J. Moir, "Subband time-frequency image texture features for robust audio surveillance," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2605–2615, 2015.
3. R. V. Sharan and T. J. Moir, "Noise robust audio surveillance using reduced spectrogram image feature and one-against-all SVM," *Neurocomputing*, vol. 158, pp. 90–99, 2015.

Conference Proceedings

1. R. V. Sharan and T. J. Moir, "Cochleagram image feature for improved robustness in sound recognition," in *Proceedings of the IEEE International Conference on Digital Signal Processing (DSP)*, Singapore, 2015, pp. 441–444.
2. R. V. Sharan and T. J. Moir, " Subband spectral histogram feature for improved sound recognition in low SNR conditions," in *Proceedings of the IEEE International Conference on Digital Signal Processing (DSP)*, Singapore, 2015, pp. 432–435.
3. R. V. Sharan and T. J. Moir, "Robust audio surveillance using spectrogram image texture feature," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia, 2015, pp. 1956–1960.
4. R. V. Sharan and T. J. Moir, "Audio surveillance under noisy conditions using time-frequency image feature," in *Proceedings of the 19th International Conference on Digital Signal Processing (DSP 2014)*, Hong Kong, 2014, pp. 130–135.

5. R. V. Sharan and T. J. Moir, "Comparison of multiclass SVM classification techniques in an audio surveillance application under mismatched conditions," in *Proceedings of the 19th International Conference on Digital Signal Processing (DSP 2014)*, Hong Kong, 2014, pp. 83–88.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
1-SVM	One-Class SVM
ADAG	Adaptive Directed Acyclic Graph
ANN	Artificial Neural Network
ASR	Automatic Speech Recognition
BBRBM	Bernoulli-Bernoulli Restricted Boltzmann Machine
BP	Basic Pursuit
BW	Bandwidth
CD	Contrastive Divergence
CIF	Cochleagram Image Feature
CITF	Cochleagram Image Texture Feature
CS	Cepstral Scaling
CMVN	Cepstral Mean and Variance Normalization
DCT	Discrete Cosine Transform
DDAG	Decision Directed Acyclic Graph
DFB	Distance-From-Boundary
DFT	Discrete Fourier Transform
DNN	Deep Neural Networks
DTW	Dynamic Time Warping
EER	Equal Error Rate
ER	Error Rate
ERB	Equivalent Rectangular Bandwidth
FSR	First-to-Second Ratio
GBRBM	Gaussian-Bernoulli Restricted Boltzmann Machine
GLCM	Gray-Level Co-occurrence Matrix
GMM	Gaussian Mixture Model
GTCC	Gammatone Cepstral Coefficients
HMM	Hidden Markov Model
*k*NN	*k*-Nearest Neighbor
MFCC	Mel Frequency Cepstral Coefficients
MP	Matching Pursuit

NFL	Nearest Feature Line
NN	Nearest Neighbor
OAA	One-Against-All
OAO	One-Against-One
PNN	Probabilistic Neural Network
PSD	Power Spectral Density
RBF	Radial Basis Function
RBM	Restricted Boltzmann Machine
RCIF	Reduced Cochleagram Image Feature
RSIF	Reduced Spectrogram Image Feature
SBE	Subband Energy
SC	Spectral Centroid
SER	Sound Event Recognition
SF	Spectral Flux
SIF	Spectrogram Image Feature
SITF	Spectrogram Image Texture Feature
SNR	Signal-to-Noise Ratio
SR	Spectral Roll-Off
SSC	Subband Spectral Centroid
STE	Short-Time Energy
SVM	Support Vector Machine
ZCR	Zero-Crossing Rate

Nomenclature

Table 1: List of symbols used in feature extraction

Symbol	Description
F_s	Sampling frequency of sound signal
N	Window length
$X(k,t)$	k^{th} harmonic in t^{th} frame for frequency $f(k) = kF_s/N$
$x(n)$	Time-domain signal
$w(n)$	Window function
$\eta(n)$	Noise signal
P_x	Average signal power
P_η	Average noise power
x_{rms}	Signal amplitude in root mean square
η_{rms}	Noise amplitude in root mean square
Z	Length of sound and noise signal
X	Desired SNR in dB
ψ_{rms}	Root mean square value of noise based on X and x_{rms}
η_x	Scaled noise signal based on ψ_{rms}
x_η	Noise manipulated signal
f_0	Half sampling frequency or Nyquist frequency
f_{sl}	Lower bound of a subband
f_{sh}	Upper bound of a subband
f_{sc}	Frequency spectral centroid
ε	Empirical constant
f_{Hz}	Frequency in Hz
f_{Mel}	Frequency in Mel
f_c	Centre frequency
m	Filter index
f_l	Minimum cut-off frequency
f_h	Maximum cut-off frequency
M_1	Total number of mel-filters
f_x	Mel-filter cut-off frequencies

$V(m,k)$	Mel-filter normalized frequency response
$E(m,t)$	Mel-filter bank energy
i	Cepstral coefficient index
l	Order of the cepstrum
c	Cepstral coefficient
g	Impulse response of gammatone filter
A	Amplitude
j	Order of the gammatone filter
W	Bandwidth of the gammatone filter
ϕ	Phase
r	Time
f_{ERB}	Frequency in ERB
Q_{ear}	Asymptotic filter quality at high frequencies
W_{min}	Minimum bandwidth for low frequency channels
M_2	Total number of gammatone filters
s	Step factor
c_{Δ}	Delta coefficient
$c_{\Delta-\Delta}$	Delta-delta coefficient
$\hat{c}$	Cepstral coefficients after scaling
S_{linear}	Linear DFT values
S_{log}	Log DFT values
$I(k,t)$	Normalized spectrogram image intensity values
μ_v	v^{th} central moment
K	Sample size or the number of pixels in a block of image
μ	Mean intensity value in a block of image
B	Number of blocks along rows and columns of image
μ_{Rb}, σ_{Rb}	Mean and standard deviation values for blocks in b^{th} row
μ_{Cb}, σ_{Cb}	Mean and standard deviation values for blocks in b^{th} column
$P(i,j)$	GLCM for element (i,j)
d_t	Offset in x direction
d_k	Offset in y direction
N_t	Number of pixels in the x direction in image I
N_k	Number of pixels in the y direction in image I
N_g	Number of quantized gray levels

N_b	Number of subbands
$\hat{x}(n)$	Gammatone filtered signal
$C(m,t)$	m^{th} harmonic in t^{th} frame for centre frequency f_{cm}

Table 2: List of symbols used in classifiers

Symbol	Description
$\mathbf{p}, \mathbf{q}$	Any two given feature vectors
l	Number of training samples
$\mathbf{x}_i$	Feature vector for the i^{th} training sample
d	Feature vector dimension
y_i	Class label of $\mathbf{x}_i$
$\mathbf{w}$	Normal vector to the hyperplane
b	Constant
α, β	Lagrange multipliers
ξ_i	Non-negative slack variables
$\sum_i \xi_i$	Penalty function
T	Penalty/tuning parameter to balance margin and training error
$\mathbf{z}$	Higher dimensional space for mapping $\mathbf{x}$
$\phi(\mathbf{x})$	Nonlinear mapping for $\mathbf{x}$
$K(\mathbf{x}_i, \mathbf{x}_j)$	Nonlinear kernel function for inner product of $\phi(\mathbf{x}_i)$, $\phi(\mathbf{x}_j)$
r	Degree of the polynomial
σ	Width of the Gaussian function
a_1	Scale parameter for multilayer perception
a_2	Offset parameter for multilayer perception
$f(\mathbf{x})$	SVM classifier output function
M	Number of classes
N_p	Number of nodes in the p^{th} layer for ADAG-SVM
$L_p(q)$	Output of the q^{th} node in the p^{th} layer for ADAG-SVM
P	Number of layers for ADAG-SVM
L	Number of layers for DNN
$\mathbf{v}$	Vector of visible nodes in DNN
V	Number of visible nodes in DNN
$\mathbf{h}$	Vector of hidden nodes in DNN
H	Number of hidden nodes in DNN

$E(\mathbf{v}, \mathbf{h})$	Energy functions of the RBM structures
w_{ji}	Weight between the i^{th} visible unit and the j^{th} hidden unit
b_i^v, b_j^h	Real valued biases
θ_{bb}	BBRBM model parameter
$\mathbf{W}$	Weight matrix
$\mathbf{b^h}, \mathbf{b^v}$	Bias matrices
$p(\mathbf{v}, \mathbf{h}; \theta)$	Joint probability associated with configuration $(\mathbf{v}, \mathbf{h})$
Y	Partition function given as $Y = \sum_v \sum_h e^{\{-E(\mathbf{v},\mathbf{h};\theta)\}}$
C	Cross entropy cost function

Chapter 1

Introduction

1.1 Overview of a Sound Event Recognition System

Any given environment generally contains a number of different sounds. In early literature, these sounds were often divided into *speech* and *non-speech*. The task of non-speech sound classification is now more commonly known as sound event recognition (SER). It is also referred as automatic sound recognition and acoustic event detection in some contexts. While research in automatic speech recognition (ASR) has received significant attention over the past few decades, research in SER only seems to have intensified over the past two decades or so.

A SER system aims to recognize sounds automatically using signal processing and machine learning techniques. It is essentially a pattern recognition problem and being a relatively new area of research, most of the techniques involved are inspired from other pattern recognition problems, ASR in particular. This is especially true when it comes to the selection of features and classifiers. While traditional methods can yield decent performance in clean or noise free conditions, it is always a challenge to achieve robust SER, that is, obtaining better recognition rate in the presence of noise. For the purpose of this work, noise is defined as any unwanted continuous signal such as people chattering in the background or TV playing.

The techniques for robust SER could be classified into three categories:

- pre-process the noisy sound signal to obtain a better estimation of the clean sound signal,
- robust feature representation of the sound signal, and

- adapt acoustic model parameters to match the noisy sound signal.

While the above techniques have merits in their own rights and a combination of these approaches is also a possibility, the interests in this research mainly lies in finding noise robust features to achieve better classification performance in SER in the presence of noise. The general technique in the many applications of SER are same and this work uses an audio surveillance application to test the developed statistical pattern recognition techniques.

An overview of a statistical pattern classifier adopted in most SER systems is given in Figure 1.1. The three key steps in implementing an automatic SER system are *signal preprocessing*, *feature extraction*, and *classification*. Signal preprocessing aims to prepare the sound signal for feature extraction. Due to limitations in digital signal processing hardware, a signal is often divided into smaller *frames*, typically in the range of 10-30 ms, and a *window* function is applied to smooth the signal for further analysis. While ASR systems typically use a sampling frequency of 8000 Hz or lower, SER systems generally employ a sampling frequency of 8000 Hz or higher, common values are 16000 Hz, 22050 Hz, and 44100 Hz, largely depending on the frequency bands of the sound signals in the database. Depending on the sampling frequency of the signal, a frame size of 256, 512, or 1024 samples are normally chosen with some degree of overlap between adjacent frames, such as 25% or 50%, to prevent loss of information around the edges of the window.

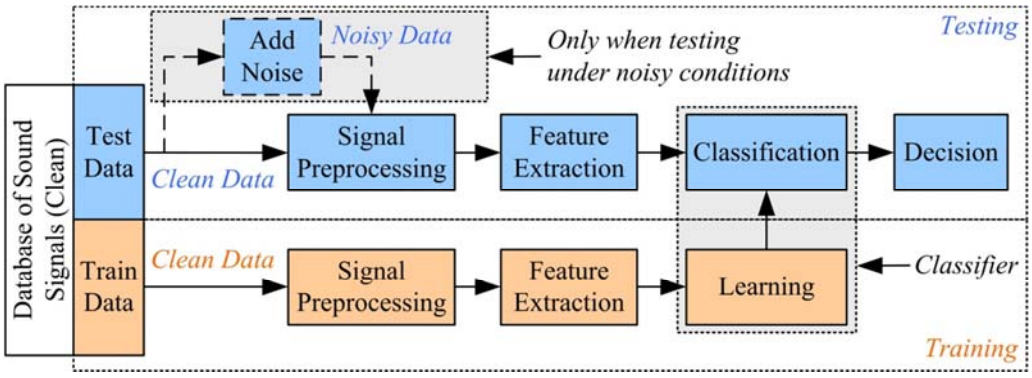


Figure 1.1: Model of a typical statistical pattern classifier employed in SER systems

Inherent features are then extracted from the signals and the input signal is represented by a *feature vector* in a much simpler and condensed form, which is referred as *feature extraction*. The time-domain signal is often transformed to frequency-domain or time-frequency domain for this purpose. Based on a set of training data containing observations whose classes are known, the task of the *classifier* is then to assign unknown observations to one of the classes. Sometimes noise is also added to the otherwise clean test signals to determine the robustness of the features and classifiers employed for the task. In addition, multi-conditional training can be employed to improve the classification performance in the presence of noise whereby noise manipulated signals are included in the training samples. However, this process can significantly increase the training time, depending on how many noise environments and noise levels are included during multi-conditional training. Multi-conditional training also makes the classifier noise dependent which means it has to be retrained for every new noise environment. As such, most work instead choose to focus on finding noise robust features and classification methods to achieve a noise independent SER system.

Features and classifiers from ASR systems are often employed in SER systems. While most of the traditional features continue to be used today, they are often complemented with new features for improved performance. A thorough review of features for audio classification is provided in [1] where the features are distinguished based on its domain which can be summarized as:

- *Temporal domain* – based on the aspect of the signal the feature represents such as amplitude, power, and zero-crossing.
- *Frequency domain* – which can be further divided into *perceptual features*, which have a semantic meaning to the human listener, and *physical features*, which give description in terms of mathematical, statistical, and physical properties of the audio signal.
- *Cepstral features* – approximate the spectral envelope.
- *Modulation frequency features* – provide information on long-term amplitude or frequency variation of the signal.
- *Eigen domain features* – representing long-term information contained in sound segments with duration of several seconds.

- *Phase space features* – capture information orthogonal to features originating from linear models.

Time domain, frequency domain, and cepstral features are by far the most commonly used features in SER systems.

In addition, the commonly used classifiers are *k*-nearest neighbor (*k*NN), Gaussian mixture model (GMM), hidden Markov model (HMM), artificial neural networks (ANN), and support vector machines (SVMs), which have been well defined in many literature. While all of these continue to be used today, modifications and hybrid classification algorithms have been proposed over the years. Also, deep learning methods, such as deep neural networks (DNN), have gained significant attention in various pattern recognition problems in recent years.

The classification performance of a SER system is mostly reported using *classification accuracy* which can be given in percentage as number of correctly classified test samples divided by the total number of test samples. The *error rate* (ER) can also be used for this purpose which can be stated as the number of misclassified test samples divided by the total number of test samples.

1.2 Motivation

Cepstral features, mel-frequency cepstral coefficients (MFCCs), in particular, have been the traditional feature choice in the many applications of SER. MFCCs have been shown to be effective in structured environments but its classification performance has been shown to be poor in the presence of noise [2]. However, features extracted from the time-frequency image, or spectrogram image, of speech or sound signals have proved effective in the presence of noise [2, 3]. In [3], spectral subband centroids (SSCs) are used as supplementary features to achieve improvement in classification accuracy in the presence of noise in ASR. In another ASR application, the dominant frequency information is captured using subband spectral centroid histograms (SSCHs) and the proposed feature was seen to be more robust than MFCCs in the presence of additive background noise [4]. For robust SER in [2], the spectrogram image is divided into multiple blocks and central

moments are computed in each block which forms the feature vector, referred as the spectrogram image feature (SIF).

Every sound signal produces a unique texture which can be visualized using a spectrogram image. The intensity values in the spectrogram image represent two-dimensional time-frequency information, that is, reveal the dominant frequency components against time. Features which capture this information can improve the recognition rate in the presence of background noise provided the noise spectrum does not contain strong spectral peaks to significantly corrupt the dominant sound signal components. Such texture classification tasks are common in image processing applications. This provides the motivation to develop novel techniques for SER which analyze the texture in the time-frequency image of the sound signal to potentially achieve robust performance in the presence of background noise in the sound signal.

1.3 Contributions

The aim of this research is to improve the sound recognition rate in the presence of environmental noise using time-frequency image analysis of the sound signal and deep learning methods. With the availability of a suitable database, an audio surveillance application is considered in this work. However, since the general approach to most SER applications is same, the techniques proposed in this work could be adapted to other applications.

In addition, for the problem of audio surveillance considered in this work, the choice of sound and noise databases is similar to [5], which is one of the most comprehensive piece of work in this area. A total of 10 classes are selected in this work to show the robustness of the proposed techniques. This is more than most other work in the area of audio surveillance such as seven classes in [6-8], and nine classes in [5]. It can generally be said that the classification accuracy decreases with an increase in the number of classes as summarized in [9] in relation to the problem of environmental sound recognition. Also, similar to [5], each sound class has multiple subclasses with interclass similarity and intraclass diversity, increasing the complexity of the problem.

The focus in this work is on two groups of features: cepstral features and time-frequency image features. Two cepstral features are considered in this work: MFCCs and gammatone cepstral coefficients (GTCCs). These two features form the reference features, referred as baseline features. Unlike [10], a similar work, this study also evaluates the performance of the cepstral features in the presence of noise. In addition, there are various parameters which can have an effect of the performance of the cepstral features. In trying to determine the optimal classification accuracy, various experiments are performed such as with the inclusion of the derivatives of the cepstral coefficients, different filter bank bandwidths, and different number of filters. Also, three different filter models are considered for GTCCs.

Furthermore, in various literature on ASR [11-13] and SER [2], the performance of cepstral features, MFCCs have been covered in particular, have been shown to be poor in the presence of noise. One reason for this is the log spectrum compression used in conventional cepstral analysis which is sensitive to noise [12]. Root cepstrum, which essentially means raising the filter bank energy values to power in the range 0 to 1 before computing the cepstral coefficients, has been proposed in [13] to improve the robustness of MFCCs in ASR. In this work, the concept of root compression is applied to MFCCs but in an application of SER and its applicability to GTCCs is also explored.

The third and final baseline feature is the SIF, derived from the spectrogram image of the sound signal. In addition, two baseline classifiers are considered in this work, *k*NN and SVM.

Research in SER has formed part of a number of PhD research around the world, as seen in [2, 5, 9], and the major contributions from this research are discussed below.

Applications and Advancements in Sound Event Recognition [14]

Research in ASR has been going on for many years now and there are many literature review papers which document the progress in ASR systems. However, the same is not true for SER. As such, this study provides a comprehensive review on the applications and advancements in SER, mainly over the last two decades.

Reduced Spectrogram Image Feature [15, 16]

In [2], the SIF was determined to be more noise robust than MFCCs but at the expense of a higher feature dimension. This work proposes a method to reduce the SIF dimension using the mean and standard deviation values of the extracted features without compromising the classification performance. This is referred as the reduced spectrogram image feature (RSIF). With the RSIF, a feature dimension same as the cepstral features was achieved, which was 2.25 times smaller than the SIF. In addition the classification performance was determined to be better than the SIF.

Spectrogram Image Texture Feature [17, 18]

Here, a new feature is proposed which is based on the image texture analysis technique of gray-level co-occurrence matrix (GLCM), also known as gray-tone spatial dependence matrix [19]. However, the GLCM technique of texture analysis is applied to sound signal spectrogram images for classification of sounds in an audio surveillance application. Also, instead of extracting textural descriptors from the GLCM, as is the norm, it is proposed to concatenate the columns of the matrix to form the feature vector for a sound signal. This is referred as the spectrogram image texture feature (SITF). Unlike in [20, 21], performance evaluation of this analysis technique is also carried under noisy conditions. In addition, texture analysis is performed in subbands, similar to the zoning technique utilized in [21]. This essentially divides the spectrogram image into horizontal sections of different frequency bands. GLCM analysis is performed independently in each frequency band and the final feature vector is a concatenation of the feature vectors from each subband. In terms of noise robustness, the SITF was seen to outperform all the baseline features considered in this work together with the RSIF and also produced the highest overall classification performance.

Cochleagram Feature Extraction [18, 22]

This work also proposes the use of cochleagram image of sound signals for feature extraction over the conventional spectrogram image. A cochleagram [23] is a variation of the spectrogram utilizing a gammatone filter, which models the human cochlea. The same features as with the spectrogram image are considered here. In

the case of cochleagram feature extraction, the spectrogram-derived features SIF, RSIF, and SITF are referred as CIF, RCIF, and CITF, respectively. For all three features, feature extraction using cochleagram was shown to give improvement in classification performance under all noise conditions with the most improved results at low signal-to-noise ratios (SNRs).

Linear GTCC + Cochleagram Image Features [18]

Feature vector combination has been shown to improve the classification performance in a number of literature. A combination of cepstral features and SSCs improved the robustness in ASR in [3]. Various feature combinations were experimented with in a similar work in [5]. Cepstral features have been shown to perform well in a noise-free environment while the strength of time-frequency image-derived features lies in noise robust performance [2]. In this work, the best performing cepstral feature was determined to be linear GTCCs and cochleagram image derived features the best time-frequency image features. Therefore, a combination of linear GTCCs and cochleagram image derived features is proposed in trying to achieve further improvement in classification performance when compared to the individual features on their own. This feature combination was shown to give further improvement in classification performance and, once again, the most improved results were at low SNRs.

Performance Evaluation of SVM and DNN Classifiers Under Noisy Conditions [16, 24]

This work also performs a comprehensive study on the performance of SVM and DNN classifiers. SVM is a binary classifier and a number of multiclass SVM classification methods have been proposed over the years. While there are a number of literature where the performance has been compared under clean conditions, this work analyzes the performance of four commonly used multiclass classification methods in the presence of noise. In addition, the performance is also compared against DNN, which, to the best of my knowledge, hasn't been used in an audio surveillance application before. The performance of the classifiers is evaluated using the classification accuracy and the training time and evaluation time with both individual and combined features. The study shows that DNN gives the best

classification performance with both individual and combined features. DNN was also shown to have the fastest evaluation time but offers the slowest training time.

Summary of Publications from Contributions

The literature review paper on the applications and advancements of SER has been published in *Neurocomputing* journal [14]. The work on RSIF and performance evaluation of multiclass SVM classification methods were presented at the *DSP 2014* conference [15, 24] and also published in *Neurocomputing* journal [16]. The work on SITF and cochleagram image feature extraction were presented at the *ICASSP 2015* [17] and *DSP 2015* [22] conferences, respectively, and also published in *IEEE Transaction on Information Forensics and Security* [18]. The work on cepstral and time-frequency image feature combination has been published in [16, 18] while the results using DNNs are currently under review.

Apart from the above, another feature developed as part of the wider research utilized the subband spectral intensity distribution, extracted from the spectrogram image of sound signals. It was referred as the spectral histogram feature (SHF). In addition, the classification performance of the SHF was improved by utilizing a mel-spectrogram, which utilizes a mel-filter, for feature extraction instead of the spectrogram image and was consequently referred as the mel-spectral histogram feature (MSHF). This work was also presented at the *DSP 2015* conference [25]. However, it has been excluded from this thesis to avoid confusion by presenting too many similar methods and also because the SHF was found to be less superior to the SITF and mel-spectrogram feature extraction was found to be less effective than cochleagram feature extraction.

1.4 Thesis Organization

The rest of this thesis is organized as follows.

Chapter 2 provides an overview of the advancements in SER as seen through some common applications of SER. Some of the less commonly known applications of SER are then discussed.

In Chapter 3, feature extraction for currently used features, which includes time and frequency domain features, cepstral features, and the SIF, is provided. Next, the

proposed features, RSIF and SITF, and the proposed time-frequency representation, cochleagram image, are presented together with their motivation.

Chapter 4 gives an overview on *k*NN and SVM classifiers and the common multiclass classification methods for SVM are also discussed. An overview of DNNs is also provided.

Experimental evaluation is provided in chapter 5. It includes information on the sound and noise databases used in this work and the experimental setup. Results are then presented using the baseline and proposed methods. Next, further analysis is carried out on the various features and classification methods.

Finally, conclusions and recommendations are provided in chapter 6.

Chapter 2

Literature Review

Initial interests in SER were mostly centered around content-based audio classification and retrieval as seen in [26-28]. The specific application of most of these early works were unclear but were eventually streamlined into applications such as music genre classification [29], musical instrument sound classification [30], and speech and non-speech recognition [31, 32].

However, applications have diversified since then with interests in areas such as audio surveillance [5] and environmental sound recognition [9]. Applications of audio surveillance systems include security monitoring in a room [33], public transport [34], and elevator [35], intruder detection in wildlife areas [36], and monitoring of elderly people, also referred as medical telemonitoring [37]. Environmental sound recognition can pose a greater challenge when compared to most other SER applications. This is because an environmental sound can comprise a number of different sound events within the environment which can be present in different combinations at any given time.

While this research looks at an audio surveillance application, a wider perspective is taken with the literature review process whereby various SER applications are considered to gauge the advancements in features and classifiers. The review is provided in the following sections and then some of the lesser known applications of SER are discussed.

2.1 Time and Frequency Domain Features

One of the early works in content-based audio classification and retrieval is by Wold et al. [26] which also found commercial success and was called Muscle Fish (www.musclefish.com). It utilized some low-level acoustical features, such as loudness, pitch, brightness or spectral centroid (SC), and bandwidth (BW), with a nearest neighbor (NN) classifier based on normalized Euclidean distance. The sound database had 409 sounds files belonging to 16 classes: *alto trombone*, *animals*, *bells*, *cello bowed*, *crowds*, *female*, *laughter*, *machines*, *male*, *oboe*, *percussion*, *telephone*, *tubular bells*, *violin bowed*, *violin pizz*, and *water*. Content-based audio retrieval has been the main application of this work with Virage Inc., BBC, and Kodak amongst its licensees [38].

Some other commonly used time and frequency domain features as seen in various literature [5, 28, 39] include zero-crossing rate (ZCR), short-time energy (STE), subband energy (SBE), spectral flux (SF), and spectral roll-off (SR). While time and frequency domain features continue to be used in SER systems, such as in audio surveillance applications [5], they are often only used as supplementary features.

2.2 Cepstral Features

Another group of features, inspired from ASR, are cepstral coefficients. A cepstrum can be defined as the inverse Fourier transform of the logarithm of the magnitude spectrum of a signal and has been widely used in the analysis of speech signals. It gives information about how the frequencies change in the spectrum and is sometimes referred as the spectrum of the spectrum.

Linear prediction cepstral coefficients (LPCCs) [40] are probably the earliest of the cepstral features. LPCCs are derived from linear prediction coefficients (LPCs) which is a technique for estimating simple speech parameters such as pitch, formants, spectra, and vocal tract. Linear prediction analysis is based on the idea that a speech sample can be approximated using a linear combination of past speech samples [41].

However, LPCCs have largely been replaced by MFCCs [42] which represent the short time power spectrum of a sound signal in a condensed form. Humans are better at differentiating small changes in pitch at low frequencies than at high frequencies. The mel-filter used in MFCCs equally spaces the frequency bands on the mel-scale [43] which more closely resembles how humans perceive sound when compared to linearly spaced cepstrums. In addition, ΔMFCCs and ΔΔMFCCs [44], also known as differential and acceleration, respectively, which provide trajectories of MFCCs over time, are often appended to MFCCs to improve the classification performance.

MFCCs have either been used as a feature on its own, as in [36], or combined with other features for improved performance as in [9, 45]. Li [27], from Microsoft Research China, extended the research of Wold et al. [26] by using MFCCs in combination with perceptual features such as total spectrum power, subband powers, brightness, bandwidth, and pitch. The audio is first classified as silent and non-silent where silent is defined as one which has the sum of the signal magnitude below a certain threshold. The mean and standard deviation of the features extracted from the non-silent frames are then concatenated to form the feature vector with normalized values. Using the same audio database as Muscle Fish, the leave-one-out test is carried out first where each of the 409 sound files are used as query but the query sound is not used as a prototype. The combination of perceptual and cepstral features gives a better classification performance than the individual features with the lowest ER of 9.78%, much better than the ER of 18.34% for the Muscle Fish system [26]. In the second test, evaluation is done using separate training and test sets, 211 files and 198 files, respectively, with the lowest ER of 9.60% using the combined features.

While MFCCs are still probably the most common feature in both ASR and SER applications, it has been shown to perform poorly in noisy conditions [2, 46]. Even with the inclusion of different features, the performance at low SNRs has generally been poor unless using multi-conditional training which requires large datasets to capture the variations in environmental conditions. In [47], power normalized cepstral coefficients (PNCCs) [48] were shown to outperform MFCCs and LPCCs under various noise conditions and noise levels. Independent component analysis (ICA) MFCCs, using FastICA algorithm to find the ICA transformation bases [49],

are proposed in [50] for recognizing home environment sounds under air-conditioner noise for home automation.

Furthermore, GTCCs are a more recent addition to the family of cepstral features. GTCCs employ a gammatone filter, a linear filter which models the frequency selectivity property of the human cochlea. The most commonly used cochlea model is that proposed by Patterson et. al. [51] which is a series of bandpass filters with the bandwidth given by equivalent rectangular bandwidth (ERB). An efficient implementation of the gammatone filter bank has been provided by Slaney [52] which has been closely followed in ASR [11] and SER [10] applications. In [11], performance of a number of front-end features, including LPCCs, MFCCs, and GTCCs, are compared under clean conditions and in the presence of white Gaussian noise at various SNRs. The results using GTCCs were seen to be better than the conventional methods. Also, a detailed analysis on MFCCs and GTCCs is performed in [10] with GTCCs determined to be more effective than MFCCs in representing the spectral characteristics of non-speech audio signals, especially at low frequencies. However, the performance wasn't evaluated in the presence of noise.

2.3 Sparse Decomposition

Sparse decomposition aims to decompose a given input signal as a linear combination of a defined number of elementary signals from a large linearly dependent collection. While there are a few algorithms for this, such as basic pursuit (BP) [53], matching pursuit (MP) seems to be the most often used in SER applications. Chu et al. [9] consider MP for environmental sound recognition. Their sound database consists of fourteen environment types, taken from BBC sound effects library [54] and the Freesound project [55], which are as follows: *inside restaurants*, *playground*, *street with traffic and pedestrians*, *train passing*, *inside moving vehicles*, *inside casinos*, *street with police car siren*, *street with ambulance sire*n, *nature-daytime*, *nature-nighttime*, *ocean waves*, *running water/stream/river*, *raining/shower*, and *thundering*.

In simple terms, MP, originally proposed by Mallat and Zhang [56], allows extraction of time-frequency features through the sparse linear expansion of a

waveform. This is done by decomposing signals using an overcomplete dictionary of functions, such as Gabor dictionary [56] as used in [9]. Some other available dictionaries include wavelets [57], wavelet packets [58], multiscale Gabor dictionaries [59], and chirplets [60]. An overcomplete dictionary ensures that the signal converges to a solution with zero residual energy and, therefore, results in the best set of functions to approximate the original representation. In [9], frequency and scale parameters are extracted from each atom as features together with the mean and standard deviation for each parameter, with five determined as the optimal number of atoms. A combination of MFCCs and MP features produced the highest classification accuracy at 83.9% using GMM classification.

Interestingly, a listening test was also given to 18 individuals with an overall accuracy of 77%, 82%, and 85% for an audio clip of duration 2, 4, and 6 seconds, respectively. The confidence level of the individuals were also measured with each answer which showed direct correlation with the accuracy. Potential short falls in the listening test, such as short duration of clips, were discussed against the results in [61] where listening tests produced better results than the automatic SER system.

Some other applications of MP include note detection in musical recordings [62], music genre recognition [63], and in automatic classification of time-varying warning signals from an acoustic monitoring system to indicate potential catastrophic structural failures of reinforced concrete structures [64].

Unlike [9], which uses overcomplete dictionaries, MP for signal approximation with sparse optimization method [65] is used for drum sound classification in [66]. Data samples from ENST database [67] and RWC Music Database: Musical Instrument database [68] are used and the following features are considered: MP features using a sparse coding dictionary (SC-MP), MP features using a gammatone dictionary (GT-MP), and timbre descriptors (TD). Apart from the three individual feature sets, the combination of TD with SC-MP and GT-MP is also considered. Results are compared under clean conditions and at -10dB, 0dB, 10dB, and 20dB SNR with the addition of white Gaussian noise. When trained with clean samples only, the overall performance of the MP features was much better than TD features. While all the features gave comparable results under clean conditions, MP features performed much better under noisy conditions, except at -10dB and 0dB SNR where all features gave poor results. The addition of MP features to TD and multi-conditional

training for TD improved its classification accuracy but the overall performance of the individual MP features was still better.

2.4 Time-Frequency Image Features

There are also some literature which use the unique approach of extracting features from the time-frequency image of the sound signal. Spectrogram images of a sound signal were used for feature extraction in a hearing aid application by Abe et al. [69]. While more than thirty features were extracted, eleven features were chosen through correlation analysis for classifying four classes: *speech*, *speech in noise*, *noise*, and *classical music*. The original image is in grayscale but binary images are also created for feature extraction. Five features are firstly used to classify between *classical music* and *the others*. *The others* is then classified as *speech*, *speech in noise*, and *noise* using the remaining six features.

In addition, Dennis et al. [2] extract central moments as features from the spectrogram image of sound signals, referred as the SIF, for sound event recognition which was shown to produce relatively good results in noisy environments. For experimentation, 60 sound categories, taken from the Real World Computing Partnership (RWCP) Sound Scene database in Real Acoustic Environment [70], are used to give a selection of *collision*, *action*, and *characteristics* sounds. Each class has 80 files of which 50 files are randomly selected for training and 30 files are used for testing. Four noise types, *speech babble*, *destroyer room control*, *factory floor 1*, and *jet cockpit 1*, from NOISEX-92 database [71] are added at 20dB, 10dB, and 0dB SNRs to test the robustness of the system. While MFCCs were seen to produce better results under clean conditions, the results using the SIF were much better at low SNRs. The best results at 0dB SNR were between 74-77% for the four noise types using the SIF with HMM classification, implemented using the HTK toolkit [72].

Furthermore, in some literature, the GLCM, an image processing based texture analysis technique, has been extended to texture analysis of sound signal time-frequency images. GLCM gives the spatial relationship of pixels in an image and Costa et al. [21] used it for texture classification of spectrogram images for music genre recognition. Their audio database consists of 900 music pieces from 10 music

genres taken from the Latin music database [73]. The audio signal is first converted to a spectrogram using time decomposition [73] and the GLCM texture descriptors are extracted as features using a zoning technique, that is, the spectrogram image is divided into horizontal sections, with a total of 10 zones, and analysis is carried out in each zone. Due to the non-uniform nature of the sound signal spectrograms, this local feature extraction technique was shown to give higher results than global features. The following seven features are extracted from the GLCM from the fourteen textural descriptors proposed in [19]: entropy, correlation, homogeneity, third order momentum, maximum likelihood, contrast, and energy.

The results are compared against those in [74] which takes an instance-based approach with feature vectors represented by short-term, low-level characteristics of the music audio signal. Only a marginal increase is seen in the average classification accuracy, increasing from 59.6% to 60.1%, but results showed an improvement of about 7% with a combination of the two methods.

The GLCM method of image texture analysis using the fourteen textural descriptors of [19], a subset of these features, or with other textural descriptors has been employed in various other applications. These include insect recognition [75], fabric surface roughness evaluation [76], and urban and agricultural land classification [77]. It has also been applied for diagnosis of abdominal tumors using texture classification of ultrasound images [78] and mammogram texture classification for breast cancer detection [79]. In a face recognition problem [20], however, instead of extracting features from the GLCM, the matrix values itself are used to form the feature vector. This approach was generally shown to give significantly better results than using the combined fourteen textural descriptors as features.

Moreover, while the spectrogram image is the most commonly used representation in time-frequency analysis of sound signals, it may not be the best choice depending on the application. Short-time Fourier transform (STFT) is a commonly used method for spectrogram image formation where the signal is divided into short duration frames and discrete Fourier transform (DFT) is applied to the windowed frames. The spectrum values from each frame are stacked side-by-side to form the spectrogram image. The spectrogram image gives dominant frequency information against time and the frequency components are equally spaced along the vertical with constant bandwidth. However, most sound signals hold greater frequency components in the

lower frequency range and, therefore, the information in these frequency bands are not fully revealed in this time-frequency representation.

Wavelet transform [80] also provides a time-frequency representation of a signal and has advantage over Fourier transform in that it provides better time and frequency localization. Nilufar et al. [81] use wavelet packet decomposition [82], an extension of wavelet transform that includes more signal filters, for robust *speech* and *music* discrimination. This technique is applied to the spectrogram to transform it into different subbands containing texture information. Multiple kernel learning (MKL) [83] is used to select the optimal subbands for discriminating the two classes.

A cochleagram is another variation of the spectrogram which uses a gammatone filter, as used for computing GTCCs, and is sometimes referred as a gammatone-spectrogram. A gammatone filter offers more frequency components in the lower frequency range with narrow bandwidths and fewer frequency components in the higher frequency range with wide bandwidths. This makes the corresponding time-frequency representation more suitable for feature extraction. Time-frequency analysis and feature extraction using cochleagram images have a number of applications in areas of signal processing and pattern recognition. For example, features were extracted from cochleagram images in [84] in trying to improve the robustness in ASR. In [85], cochleagram image features outperform a combination of common acoustic features in voice activity detection. Similar approach is also taken in [86] for audio separation purposes.

2.5 Support Vector Machines

2.5.1 Binary Support Vector Machines

SVM is a statistical learning classifier developed for binary classification. The initial SVM was a linear classifier proposed by Vapnik and Lerner in 1963 [87]. This was extended to nonlinear datasets by Boser, Guyon, and Vapnik in 1992 [88] and has gained widespread attention since the late '90s, around the same time research in SER was generating interests. Being a binary classifier, a number of techniques have been proposed for multiclass classification. The most common technique is to

reduce the multiclass classification problem into multiple binary classification problems. Four commonly used methods based on this technique are one-against-all (OAA) [89], one-against-one (OAO) [90], decision directed acyclic graph (DDAG) [91], and adaptive directed acyclic graph (ADAG) [92].

Guo and Li [28] used the Muscle Fish database and similar features as in [27], that is, cepstral and perceptual features. However, a new metric called distance-from-boundary (DFB) is proposed for audio retrieval using SVMs to learn the boundaries. SVM, with a bottom-up binary tree structure, similar to ADAG method, is proposed to reduce the number of comparison during testing. Exponential RBF is used as the kernel function which was found to give better results than polynomial, Gaussian RBF, and multilayer perception. Using the same feature vector formation technique as [27], SVM performed better than NN, *k*NN, and NC (nearest center) classifiers. The lowest error rate is 11.00% for the leave-one-out test but 8.08% with separate training and test sets.

In another similar work, Lu et al. [39] consider five audio classes: *silence*, *music*, *background sound*, *pure speech*, and *non-pure speech*. SVM, with a Gaussian radial basis function (RBF) kernel, is used for classification with a bottom-up binary tree for multiclass SVM classification, similar to [28]. For experimentation, a database with 2600 audio clips is created with a total duration of about 4 hours obtained from TV programs, internet, audio, and music CDs. When tested under different testing units (durations), in general, *k*NN classifier gave higher results than GMM while the SVM classifier always outperformed *k*NN and GMM classifiers.

In some other applications of SVMs in SER systems, in [66], OAO-SVM with RBF kernel, implemented using LIBSVM [93], is used for drum sound classification. In [94], multi-layer perceptron (MLP) neural network, trained using the Levenberg-Marquardt (LM) [95] back-propagation algorithm, and SVM, only the polynomial kernel was considered, are experimented for classification for automatic ontology generation for musical instruments. The average classification accuracy for the MLP classifier were 76.0% and 46.7% for *solo music* and *isolated notes*, respectively, which increased to 83.0% and 86.3% with SVM classification.

In addition, there are various other pattern recognition problems where the multiclass SVM classification methods have been compared. Hsu and Lin [96]

compare the performance of OAA, OAO, DDAG and two altogether methods, an approach for multiclass problems by solving a single optimization problem, on large classification problems. They conclude OAO and DDAG as being more suitable for practical use. A similar comparison is done by Seo [97] using OAA, OAO, DDAG together with the approach given by Weston and Watkins [98] and Crammer and Singer [99] for a face recognition application. While OAO was found to give marginally better results than DDAG, DDAG is suggested due to its low computational cost.

2.5.2 One-Class Support Vector Machines

One-Class Support Vector Machines (1-SVMs), proposed by Schölkopf et al. [100], is a modification of binary SVMs to solve one-class classification problem. Here, the feature is transformed by the kernel and the origin is treated as the second class. 1-SVM essentially separates the feature data points from the origin and maximizes the distance from the hyperplane to the origin.

1-SVM is more suited with high dimensional feature vectors. As such, unlike most other work where mean and standard deviation values of the extracted features across all frames are concatenated to form the feature vector, a slightly different approach to feature data representation is taken by Rabaoui et al. [5], which is also one of the most comprehensive piece of work in an audio surveillance application. The overall feature data for the sound signal is divided into three portions: 30%, 40%, and 30% of the total number of frames. Mean value of the data across each dimension from each portion are concatenated to form the feature vector which results in a feature dimension which is 1.5 times longer than the conventional technique.

Various features were considered in this work on a database that consists of 1015 sounds files belonging to 9 classes taken from the RWCP Sound Scene database [70] and [101]: *human screams*, *gunshots*, *glass breaking*, *explosions*, *door slams*, *dog barks*, *phone rings*, *children voices*, and *machines*. Noise signals were added from the NOISEX-92 database [71] and some hand recorded signals were also used. The choice of the sound database has some similarity to other audio surveillance applications such as [7, 8, 35, 102, 103]. The classification accuracy of 1-SVM was generally higher than HMM, OAA-SVM, and OAO-SVM classification methods

when tested at various SNRs with a number of individual and combined features. A maximum classification accuracy of 96.89% was achieved under clean conditions and 93.33%, 89.22%, 82.80%, and 72.89% with the addition of noise at 20dB, 10dB, 5dB, and 0dB SNRs, respectively, with the best performing feature set with 70% of clean data used for training and the remaining for testing.

One drawback of the approach in [5] is that different combination of features were shown to produce best results under different conditions. For example, MFCCs are used to form the final feature vector under clean conditions but MFCCs are not used under noisy conditions. The implementation of such a system in real-time can be complex since it requires prior knowledge on whether there is noise present in the sound signal before selecting the best set of features or else sacrificing the classification performance using a feature set that gives the best overall performance.

2.6 Deep Neural Networks

While SVMs have seen an increased usage in SER systems, a new machine learning algorithm called deep learning is generating a lot of interest in ASR. Deep learning aims to learn high-level representations of data through a hierarchy of intermediate representations, such as deep neural networks (DNN) [104]. It has been used for acoustic modeling by research groups at University of Toronto, Microsoft Research, Google, and IBM Research, amongst others, and shown to outperform a number of classification methods [105].

McLoughlin et al. [106] compare the classification performance of DNNs against SVMs for sound event recognition. Classification performance was evaluated on three feature sets, MFCCs, SAI [107], and SIF. DNNs generally gave significantly higher overall classification accuracy with the best overall performance achieved using the SIF. With multi-conditional training, for example, the average classification accuracy using SVMs is 88.55% but 92.58% with DNNs. Similar conclusions were also drawn with MFCCs and SAI features.

2.7 Summary of Advancements and Proposed Methods

Table 2.1 summarizes some key works in SER and highlights the advancements in features and classifiers using the work by Wold et al. [26] as basis.

MFCCs have evolved as a baseline feature in many SER systems. However, it is often supplemented with other features such as perceptual features [27, 28] and MP-based features [108] for improved classification performance. The performance of MFCCs has been shown to be poor in the presence of noise and various modifications have been proposed for a more noise robust performance. GTCCs are one of the recent of the cepstral coefficients and shown to be more noise robust than MFCCs in ASR [11].

However, the strength of the cepstral features lies in classifying noise-free signals and even with the various proposed improvements, the performance at low SNRs has been poor. The use of time-frequency image derived features has been shown to be effective in the presence on noise in ASR and two such features are SSCs [3] and SSCHs [4]. Similarly, the SIF was proposed for SER in [2]. In [2, 4], the time-frequency image features were seen to be more noise robust than MFCCs. Also, in [109], the SIF was shown to be significantly more noise robust than the feature combination of MFCCs and MP proposed in [9].

This work utilizes MFCCs, GTCCs, and the SIF as baseline features. Next, a technique is proposed to reduce the feature dimension of the SIF without sacrificing the classification performance, which is referred as the RSIF. Also, a new spectrogram derived feature is proposed which performs subband texture analysis using the image texture analysis technique of GLCM and is referred as the SITF. In addition, feature extraction using cochleagram image, a variation of the spectrogram image utilizing a gammatone filter, is proposed, a technique which has been seen to be effective in ASR applications [84-86]. Finally, feature combination has been seen to improve the classification performance in a number of literature in ASR [3] and SER [5, 9, 27] and this work proposes a combination of cepstral and time-frequency image features.

Table 2.1: A summary of some key works in sound event recognition

Reference	Year	Application	Sound (Noise) Database(s)	No. of Classes (Total Files)	% Training data	Best feature(s)	Classifier	Classification Accuracy (or Error Rate)
Wold et al. [26]	1996	Content-based audio classification	Muscle Fish	16 (409)	–	Perceptual features	NN	19.07% (ER)[1]
Li [27]	2000	Content-based audio classification	Muscle Fish	16 (409)	Leave-one-out test	**MFCC** + perceptual features	NFL	9.78% (ER)
					51.59% (211/409)			9.60% (ER)
Guo and Li [28]	2003	Content-based audio classification	Muscle Fish	16 (409)	Leave-one-out test	MFCC + perceptual features	**SVM**	11.00% (ER)
					51.59% (211/409)			8.08% (ER)
Rabaoui et al. [5]	2008	Audio surveillance	RWCP Sound Scene, Leonardo Software, hand recorded (NOISEX-92, hand recorded)	9 (1015)	70%	(Multiple features[2])	**1-SVM**	Clean – 96.89% 20dB – 93.33% 10dB – 89.22% 5dB – 82.80% 0dB – 72.89%
Chu et al. [9]	2009	Environmental sound recognition	BBC Sound Effects, Freesound	14	75%	**MFCC + MP**	GMM	83.9%
Dennis et al. [2]	2011	Sound event recognition	RWCP Sound Scene (NOISEX-92)	60 (4800)	62.5%	**SIF**	HMM	[3]Clean – 87.9% [3]20dB – 88.0% [3]10dB – 87.5% [3]0dB – 75.5%
McLoughlin et al. [106]	2015	Sound event recognition	RWCP Sound Scene (NOISEX-92)	50 (4000)	62.5%	SIF	**DNN**	Clean – 96.20% 20dB – 95.80% 10dB – 94.13% 0dB – 85.47%

[1]ER as reported in [27, 28].
[2]Different combination of features were experimented with under clean and noisy conditions.
[3]Average classification accuracy value is given for the classification accuracy values reported for the four noise types.

Furthermore, SVM has been the classifier of choice in a number of SER applications and various multiclass classification methods have been experimented with. The difference in the classification accuracy between the multiclass SVM classification methods in most cases is minimal and, as such, the preference of one technique over the others is largely based on faster training and evaluation times. However, most such analysis is limited to clean conditions and it is unclear which approach is more suitable for classification under noisy conditions. In this work, the performance of the OAA, OAO, DDAG, and ADAG multiclass SVM classification methods are compared under different noise environments and SNRs. The performance of each method is evaluated using its classification accuracy and the training and evaluation times are also compared.

While SVMs have been the preferred classifier in most SER applications, DNNs have gained popularity in recent years with its superior classification performance, as demonstrated in a number of pattern recognition problems. As such, similar to [106], DNNs are also considered in this work and the performance is evaluated against SVMs with a number of individual features. In addition, the performance is compared with feature combination and the training and evaluation times are also compared.

2.8 Other Applications of Sound Event Recognition

The applications of SER are not limited to content-based audio retrieval, such as music genre and musical instrument sound classification, audio surveillance, and environmental sound recognition which have been the focus so far. Some less conventional applications of SER are discussed in the following subsections and a summary provided in Table 2.2.

2.8.1 Biometrics Identification

Similar to automatic finger print recognition, face recognition, and, more recently, vein pattern recognition systems, heart sound recognition has the potential for human identification. The use of such physiological characteristics for human identification is referred as biometrics identification. An example of such a system is given by Beritelli [110] where a database of digital heart sound recordings from 50

different people are used. MFCCs are used as features together with a feature called first-to-second ratio (FSR), power ratio of the first and second heart sounds. An equal error rate (EER) of 8.70% was achieved where EER is defined as the point where the false accept rate is equal to the false reject rate.

2.8.2 Biomedical Engineering

Automatic heart and lung sound recognition can also be used for diagnosis of disorders associated with the heart and lung, respectively. This process is often carried out manually by medical practitioners and can be subject to human error. As such, an automatic recognition system could be utilized for verification purposes. Kwak and Kwon [111] used heart sound signals for classification of cardiac disorder. MFCCs are first extracted from the heart sound signals and then the input signal is partitioned using a HMM. HMM state likelihood and murmur likelihood are then computed and combined for classification using SVM.

Furthermore, Chang and Cheng [112] study the effect of noise on lung sound recognition. Three types of lung sounds, *normal*, *wheeze*, and *crackle*, taken from the Stethographics website [113], and three noise types, *Gaussian white*, *babble*, and *car* noises, from NOISEX-92 database [71], are used to form the sound and noise databases, respectively. Three feature representations: autoregressive (AR) coefficients [114], MFCCs, and bispectrum diagonal slices (BDS) [115] are considered with dynamic time warping (DTW) [116] for classification. Such techniques have also been applied for breath sound classification [45] and snore sound detection [117].

In some other works, cough sound recognition has been applied to animals such as for identification of respiratory infections in pigs [118] and dairy calves [119]. Such technology can act as an early warning system which could help contain contagious viruses before it becomes widespread with some viruses from animals, such as swine flu, known to affect humans as well. In addition, diagnosis of disorders using SER technology extends beyond heart and lung sound recognition. An example of gastrointestinal motility monitoring system using bowel sounds, captured through abdominal surface vibrations, can be found in [120].

2.8.3 Animal Sound Recognition

Animal species recognition through analysis of their call sound is another application of SER. The benefits of such as a system are twofold. Firstly, it can be used to carry out automatic animal species recognition and monitoring replacing the laborious manual recognition and monitoring process. Secondly, it can be used for environmental monitoring since the abundance of wildlife would generally indicate a healthy environment.

For example, researchers in Brisbane, Australia, established a sensor network in the city's suburbs and forest park to study the impact of urbanization of neighboring suburbs on the ecological system, with the focus on recognition of bird species using acoustic signal analysis with MFCCs as features [121]. Frog species identification is another such application as presented in [122] where STE and ZCR are used for segmentation, MFCCs as features, and *k*NN with Euclidean distance measure for classification.

Marine mammal sound classification is another example as given in [123] for classification of 75 calls of northern resident killer whales into seven call types using cepstral features, features extracted using VOICEBOX [124]. A classification accuracy of 92% was achieved using GMM but HMM produced better results, over 95% in some cases. In addition, an illustration of classification of insect sounds using MFCCs and probabilistic neural network (PNN) can be found in [125].

SER can also be used for monitoring animal activities. In [126], automatic recognition of ingestive sounds (*bites*, *chews*, and *chewbites*) of cattle, recorded using two wireless microphones placed on the forehead of the animal, is presented for monitoring grazing behavior. Experimentation was carried out on two different pastures, *alfalfa* and *fescue*, with two heights, *tall* (24.5 ± 3.8 cm) or *short* (11.6 ± 1.9 cm), using spectral features and HMM classification, implemented using the HTK toolkit. An average recognition rate of 79.5% was achieved. An example of automatic measurement of feed intake of broiler chickens by detecting pecking sounds can be found in [127]. Another similar work but to estimate the feed consumption of giant tiger prawns by using SER for classifying feed events can be found in [128].

2.8.4 Audio-Visual Systems

While the focus so far has been on standalone applications of SER, audio and video recognition systems could also be integrated for a more holistic approach in addressing problems such as in the development of surveillance systems. Video surveillance systems have been around for many years but have limitations such as relatively expensive computation and data storage and limited field of view. A SER system could be used to complement a video-based surveillance system such as in public transports [129] and banks [6]. Audio and video recognition systems could also be combined for recognition of complex events in movies [130].

Robotics

Vision systems are also common in robotics such as for navigation purposes. Robots are often aimed at mimicking human behavior and similar to humans, acoustical information could be utilized to make a more complete description of the scene as in [131]. There is also scope for mobile robots mounted with audio and visual sensors for surveillance applications as in [132, 133]. Robotics based rescue operation is another example such as in the aftermath of an earthquake where the injured could be behind collapsed structures and audio information such as screaming or crying could be used to reach them [134].

Context Awareness

Context awareness is a computing term associated with mobile devices and aims to determine the user environment which could in turn be used to control certain internal processes. The applications for context awareness can be expanded though and an example of a social activity recognition and recommendation system using audio data gathered from mobile phone is given in [135]. However, context awareness using acoustic signal only can have limitations if the environment needs to be further classified such as indoor/outdoor or whether it is dark/bright. With a rise in handheld electronic devices equipped with audio and video sensors, such as smart phones and tablet computers, context awareness using audio and visual data is yet another application. An example of such a system is given by Choi et al. [136] with view of adding more intelligence to smart devices, with focus on smart phones. Their proposed context awareness system recognizes 10 acoustic signals: *babble*, *car*, *bag*, *music*, *noisy*, *office*, *one-talk*, *public*, *subway*, and *water*; and 4 visual

signals: *low lighting detection*, *face detection*, *indoor/outdoor detection*, and *moving detection*. The acoustic module of the system has MFCCs as features and GMM classification. The visual module consists of three detectors and a classifier, refer to [136] for details. The overall classification accuracy reported are: 98.73% for acoustic recognition, 99.27% for *low intensity detection*, 98.55% for *face detection*, 94.86% for *moving detection*, and 93.14% for *indoor/outdoor detection*. Such a system can be useful in a situation where the user cannot answer the phone, such as when driving or in a meeting, and an automatic notification could be sent to the caller and/or the ringtone muted depending on the detected activity.

2.8.5 Others

Some other applications of SER include tile wall inspection through analysis of impact sound [137], aircraft takeoff noise classification [138], helicopter type identification using rotor sound [139], identification of sound for pass-by noise test in vehicles [140], acoustic hazard detection in the form of approaching vehicles for pedestrians [141], and classification of cooking stages such as different stages of boiling water using audio and vibration signals [142].

2.8.6 Summary of Some Lesser Known Applications of SER

A summary of some these lesser known applications of SER is given in Table 2.2.

Table 2.2: A summary of some lesser known applications of sound event recognition

Reference	Application	Description	Sound Database(s)[1]	Feature(s)	Classifier(s)
Beritelli and Spadaccini [110]	Biometrics	Heart sound recognition for human identification	–	MFCC + FSR	Euclidean distance measure
Kwak and Kwon [111]	Biomedical	Heart sound classification for diagnosis of cardiac disorder	Heart Sounds and Murmurs [143]	MFCC	HMM, SVM
Lei et al. [45]		Breath sound classification for diagnosis of disorders associated with breathing	–	MFCC + perceptual features	SVM, ANN
Exadaktylos et al. [118]		Cough sound recognition in pigs	–	Power spectral density (PSD)	Euclidean distance measure
Dimoulas et al. [120]		Gastrointestinal motility monitoring using bowel sounds, captured through abdominal surface vibrations	–	Time and frequency domain features, wavelet analysis	ANN
Cai et al. [121]	Animal species recognition; sound classification; monitoring	Bird species recognition using bird calls	Backyards [144], Australian bird calls: subtropical east [145] and voices of subtropical rainforests [146], and recorded data	MFCC	ANN
Jaafar and Ramli [122]		Frog species recognition	–	MFCC	*k*NN
Brown and Smaragdis [123]		Northern resident killer whale sound classification	–	MFCC	HMM
Le-Qing [125]		Insect sound classification	United States department of agriculture [147]	MFCC	PNN [148]

Table 2.2: A summary of some lesser known applications of sound event recognition (continued)

Reference	Application	Description	Sound Database(s) [1]	Feature(s)	Classifier(s)
Milone et al. [126]	Animal species recognition; sound classification; monitoring (contd.)	Monitoring grazing behavior of cattle using ingestive sound classification	–	Spectral features	HMM
Aydin et al. [127]		Automatic measurement of feed intake of broiler chickens by detecting pecking sounds	–	PSD	(Adaptive threshold)
Yao et al. [135]	Context awareness	Context awareness for social activity recognition and recommendation using audio data gathered from mobile phone	–	MFCC, ZCR, SF, SC, BW	DTW
Tong et al. [137]	Tile Inspection	Inspection of tile wall exfoliation through analysis of impact sound	–	PSD	ANN
Márquez-Molina et al. [138]	Aircraft classification	Aircraft classification using aircraft take-off noise	–	MFCC, Octave analysis [149, 150]	ANN
Montazer et al. [139]		Helicopter type identification using rotor sound	–	Energy	RBFNN
Redel-Macías et al. [140]	Vehicle pass-by noise test	Identification of sound for pass-by noise test in vehicles	–	Spectral features	ANN
Tabacchi et al. [142]	Classification of cooking stages	Classification of cooking stages of boiling water using audio and vibration signals	–	MFCC	Parzen [151]

[1]Sound database provided only where known. Hand recorded signals were mostly used otherwise.

Chapter 3

Feature Extraction

The extraction of various features is described in this chapter. These are divided into two sections: current methods and proposed methods. The current methods include various time and frequency domain features, which are considered for feature combination, and the baseline features, cepstral coefficients and the SIF. The proposed features are described next which include the RSIF, reduced version of SIF, and the SITF, based on the GLCM method of image texture analysis. This is followed by an overview of cochleagram feature extraction and the motivation for the proposed methods.

For frequency domain analysis, the signal needs to be firstly transformed to frequency domain. For this purpose, the signal is divided into frames and DFT is applied to the windowed frames as

$$X(k,t)=\sum_{n=0}^{N-1} x(n)w(n)e^{\frac{-2\pi ikn}{N}}, \qquad k=0,1,...,N-1 \tag{3.1}$$

where N is the length of the window, $x(n)$ is the time-domain signal, $w(n)$ is the window function, and $X(k,t)$ is the k^{th} harmonic corresponding to the frequency $f(k)=kF_s/N$ for the t^{th} frame, F_s is the sampling frequency.

A Hamming window function is used in this work, similar to [2, 5], with a 50% overlap between frames to ensure that information on the edges of the window function are not lost. The Hamming window function can be given as

$$w(n)=0.54-0.46\cos\left(\frac{2\pi n}{N-1}\right),\ 0\le n\le N-1 \tag{3.2}$$

the time and frequency response of which are given in Figure 3.1.

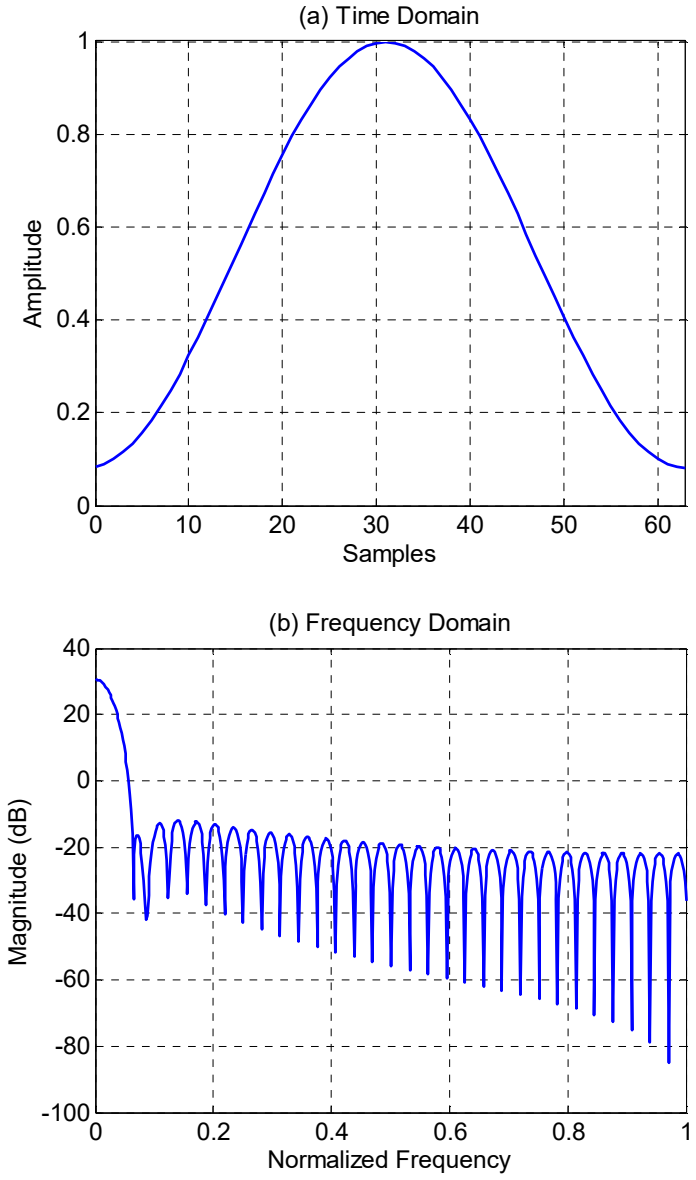


Figure 3.1: Time and frequency response of a Hamming window

All features are extracted from the sound signal under clean conditions and in the presence of noise at various SNRs. The SNR is the average power ratio between a signal $x(n)$ and the background noise $\eta(n)$ given as

$$SNR = \frac{P_x}{P_\eta} = \frac{x_{rms}^2}{\eta_{rms}^2} \tag{3.3}$$

where x_{rms} and η_{rms} are the root mean square (rms) value of the sound signal and noise signal, respectively.

It can also be expressed on the logarithmic decibel scale as

$$SNR = 10\log_{10}\left(\frac{P_x}{P_\eta}\right) = 20\log_{10}\left(\frac{x_{rms}}{\eta_{rms}}\right) \tag{3.4}$$

which is the definition used from now on in this work.

Given a signal $x(n)$ and background noise $\eta(n)$ of same length Z, if a SNR of X dB is desired, the required noise magnitude is determined as

$$\psi_{rms} = \frac{x_{rms}}{10^{0.05X}} \tag{3.5}$$

using which the scaled noise signal can be determined as

$$\eta_x(n) = \frac{\psi_{rms}}{\eta_{rms}} \times \eta(n). \tag{3.6}$$

Finally, the noise manipulated sound signal is obtained as

$$x_\eta(n) = x(n) + \eta_x(n). \tag{3.7}$$

3.1 Current Methods

3.1.1 Time and Frequency Domain Features

Zero-Crossing Rate (ZCR)

Zero-crossing rate is the number of time-domain zero-crossings within a frame and is a simple measure of the frequency content of a signal given as

$$ZCR = \frac{1}{2(N-1)} \sum_{n=1}^{N-1} \left|\text{sgn}\left[x(n+1)\right] - \text{sgn}\left[x(n)\right]\right| \tag{3.8}$$

where sgn[·] is a sign function:

$\text{sgn}[x(n)] = 1, x(n) \geq 0$, and

$\text{sgn}[x(n)] = -1, x(n) < 0$.

Short-Time Energy (STE)

Short-time energy is the total spectrum power of a frame given as

$$STE = \log \sum_{f=0}^{f_0} |X(f)|^2 \tag{3.9}$$

where $X(f)$ denotes the DFT coefficients, $|X(f)|^2$ is the power at frequency f, and f_0 is the half sampling frequency or Nyquist frequency.

Subband Energy (SBE)

Subband energy is the ratio between subband power and the total power in a frame given as

$$SBE = \frac{1}{STE} \sum_{f=f_{sl}}^{f_{sh}} |X(f)|^2 \tag{3.10}$$

where f_{sl} and f_{sh} are the lower and upper bound of a subband, respectively, with the frequency spectrum divided into four subbands: $[0, {}^{f_0}/_8], [{}^{f_0}/_8, {}^{f_0}/_4], [{}^{f_0}/_4, {}^{f_0}/_2], [{}^{f_0}/_2, f_0]$.

Spectral Centroid (SC)

Spectral centroid, also called brightness, is the frequency centroid of the spectrum or the balancing point of the spectral power distribution and is given as

$$SC = f_{sc} = \frac{\sum_{f=0}^{f_0} fX(f)}{\sum_{f=0}^{f_0} X(f)}. \tag{3.11}$$

Bandwidth (BW)

Bandwidth is the square root of the power-weighted average of the squared difference between the spectral components and frequency centroid given as

$$BW = \sqrt{\frac{\sum_{f=0}^{f_0} (f - f_{sc})^2 |X(f)|^2}{\sum_{f=0}^{f_0} |X(f)|^2}}. \tag{3.12}$$

Spectral Roll-Off (SR)

Spectral roll-off is the frequency below which a certain amount of power spectrum lies and can be determined as

$$SR = \max\left\{ F \mid \sum_{f=0}^{F} \left|X(f)\right|^2 < \varepsilon \sum_{f=0}^{f_0} \left|X(f)\right|^2 \right\} \tag{3.13}$$

where ε is an empirical constant ranged between 0 and 1 (commonly used value is 0.95) and normally half the size of the DFT is used.

3.1.2 Mel-Frequency Cepstral Coefficients

This subsection outlines the procedure for extracting MFCCs, and the following subsection for GTCCs, with reference to Figure 3.2.

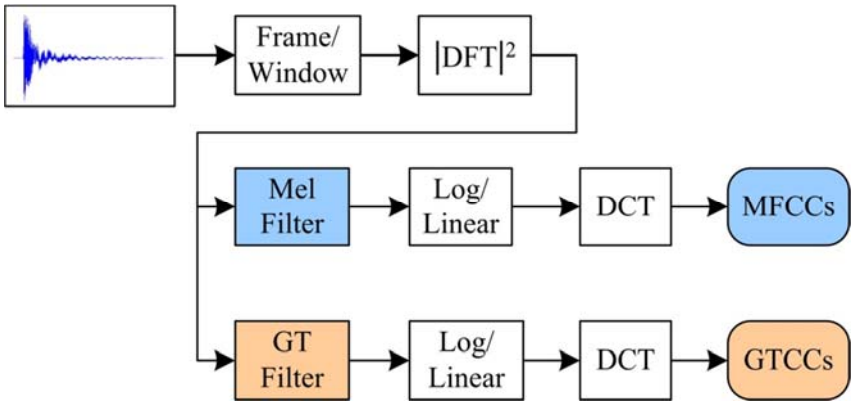


Figure 3.2: Steps in computing MFCCs and GTCCs

A key feature of MFCCs is the use of mel-filter banks or triangular bandpass filters. The filters are equally spaced on the mel-scale, a nonlinear frequency scale which more closely resembles how humans perceive sound. The conversion from frequency in Hz, f_{Hz}, to frequency in mel, f_{Mel}, can be given as [152]

$$f_{Mel} = 1127 \log\left(1 + \frac{f_{Hz}}{700}\right) \tag{3.14}$$

and the relationship is plotted in Figure 3.3 which is approximately linear below 1 kHz and logarithmic above 1 kHz.

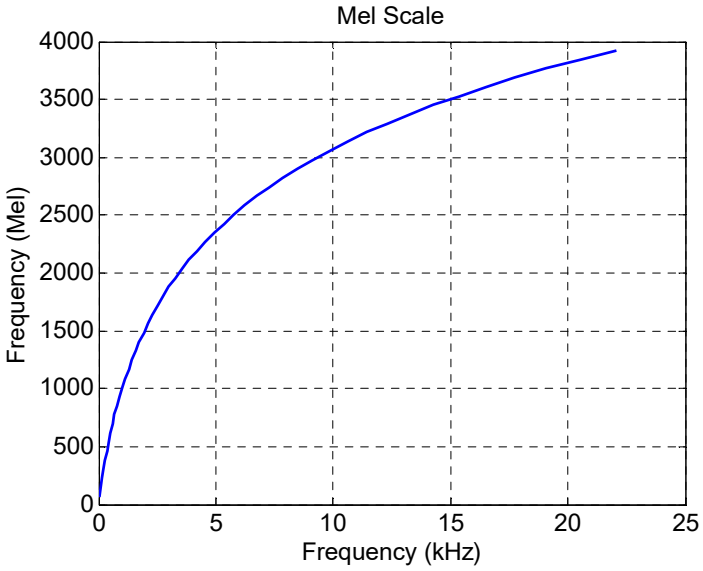


Figure 3.3: Mel scale

The center frequency for the m^{th} filter can be computed as

$$f_{cm} = f_l + \frac{m(f_h - f_l)}{M_1 + 1}, \qquad m = 1,2,...,M_1 \tag{3.15}$$

where all the frequency values are given in mel, f_l and f_h are the minimum and maximum cut-off frequencies, respectively, and M_1 is the total number of mel-filters.

The adjacent filters overlap such that the lower and upper end of a filter are located at the center frequency of the previous and next filter, respectively, while the peak of the filter is at its center frequency. The normalized frequency response can be determined as

$$V(m,k) = \begin{cases} \dfrac{f(k) - f_x(m)}{f_x(m+1) - f_x(m)}, & f_x(m) \le f(k) \le f_x(m+1) \\ \dfrac{f_x(m+2) - f(k)}{f_x(m+2) - f_x(m+1)}, & f_x(m+1) \le f(k) \le f_x(m+2) \\ 0, & Otherwise \end{cases} \tag{3.16}$$

where $k = 0,1,\dots,N/2-1$, f_x are the $M_1 + 2$ cut-off frequencies, and evaluated for $m = 1,2,\dots,M_1$. An example of a 10 channel mel filter bank is shown in Figure 3.4.

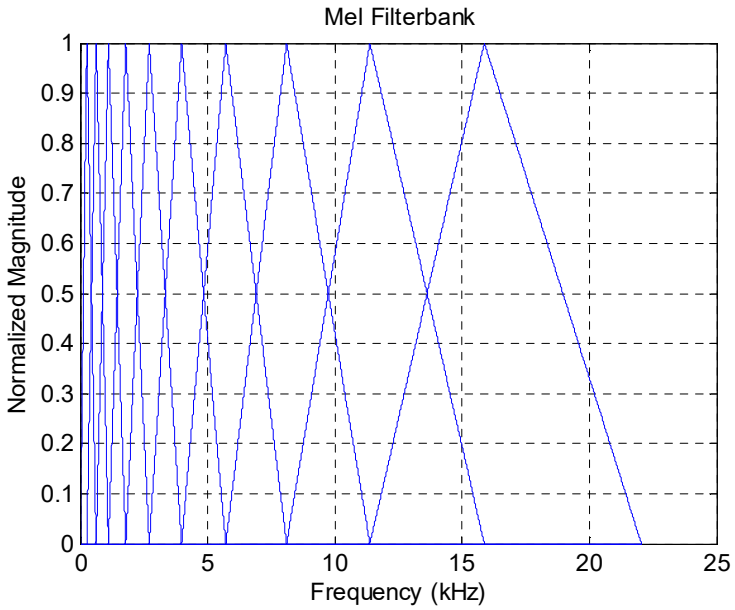


Figure 3.4: Example of a 10 channel mel filter bank

For the t^{th} frame, the output of the m^{th} filter, referred as filter bank energies, can then be determined as

$$E(m,t) = \sum_{k=0}^{\frac{N}{2}-1} V(m,k)\left|X(k,t)\right|^2, \qquad m = 1,2,...,M_1. \tag{3.17}$$

In some literature, the spectrum values are not squared in computing the filter banks energies but after experimenting with both techniques, squaring the filter bank energies was shown to give better results. The results without squaring the filter bank energies are given in [16].

The MFCCs are then obtained as the discrete cosine transform (DCT) of the log compressed filter bank energies given as

$$c(i,t) = \sqrt{\frac{2}{M_1}} \sum_{m=1}^{M_1} \log\left(E(m,t)\right) \cos\left(\frac{\pi i}{M_1}(m-0.5)\right), \qquad i = 1,2,...,l. \tag{3.18}$$

where l is the order of the cepstrum.

3.1.3 Gammatone Cepstral Coefficients

Extraction of GTCCs follows the same procedure as MFCCs except that gammatone filters are used instead of mel-filters. Gammatone filter banks are a series of bandpass filters the impulse response for which can be given as [51]

$$g(r) = Ar^{j-1}e^{-2\pi Wr}\cos(2\pi f_c r + \phi) \tag{3.19}$$

where A is the amplitude, j is the order of the filter, W is the bandwidth of the filter, f_c is the center frequency of the filter, ϕ is the phase, and r is the time.

The ERB is used to describe the bandwidth of each cochlea filter in [51]. ERB is a psychoacoustic measure of the auditory filter width at each point along the cochlea and can be given as

$$f_{c,ERB} = \left[\left(\frac{f_{c,Hz}}{Q_{ear}}\right)^p + (W_{min})^p\right]^{1/p} \tag{3.20}$$

where Q_{ear} is the asymptotic filter quality at high frequencies and W_{min} is the minimum bandwidth for low frequency channels. The bandwidth of a filter can then be approximated as $W = 1.019 \times f_{c,ERB}$. The three commonly used ERB filter models are given by Glasberg and Moore [153] ($Q_{ear} = 9.26$, $W_{min} = 24.7$, and $p = 1$), Lyon's cochlea model as given in [154] ($Q_{ear} = 8$, $W_{min} = 125$, and $p = 2$), and Greenwood [155] ($Q_{ear} = 7.23$, $W_{min} = 22.85$, and $p = 1$).

The human cochlea has thousands of hair cells which resonate at their characteristic frequency and at a certain bandwidth. In [52], the mapping between center frequency and cochlea position is determined by integrating the reciprocal of (3.20) with a step factor parameter to indicate the overlap between filters. This can then be inverted to find the mapping between filter index and center frequency which can be given as

$$f_{cm} = -Q_{ear}W_{min} + (f_h + Q_{ear}W_{min})e^{-ms/Q_{ear}} \tag{3.21}$$

where $m = 1,2,\ldots,M_2$, M_2 is the number of gammatone filters, f_h is the maximum frequency in the filter bank, and s is the step factor parameter given as

$$s = \frac{Q_{ear}}{M_2} \log\left(\frac{f_h + Q_{ear} W_{min}}{f_l + Q_{ear} W_{min}} \right) \tag{3.22}$$

where f_l is the minimum frequency in the filter bank.

A 4th order gammatone filter with four filter stages and each stage a 2nd order digital filter was used in this work as given in [52]. The gammatone filter was implemented using the Auditory Toolbox for Matlab [156]. After determining the frequency response of the gammatone filter, the steps in computing GTCCs are same as for MFCCs. An example of a 10 channel gammatone filter bank is shown in Figure 3.5. The frequency response of a gammatone filter, using Lyon's cochlear model, with a center frequency of approximately 1 kHz is shown in Figure 3.6 along with the frequency response of a mel-filter.

Delta and Delta-Delta Coefficients

The cepstral coefficients, or static coefficients, are often appended with their first and second derivatives, commonly known as delta and delta-delta coefficients, respectively. The delta coefficients can be computed as [44]

$$c_\Delta(i,t) = \frac{\sum_{d=1}^{D} d\left[c(i+d,t) - c(i-d,t) \right]}{2\sum_{d=1}^{D} d^2} \tag{3.23}$$

where $c_\Delta(i,t)$ is the i^{th} delta coefficient in the t^{th} frame and the value of D is often set to 2. The same formula can be applied to the delta coefficients to compute the delta-delta coefficients, $c_{\Delta-\Delta}(i,t)$.

Root Compression

Root compressed cepstral coefficients are computed similar to the conventional method but root compression is applied to the filter bank energies instead of log compression. Root compressed cepstral coefficients can be determined as [13]

$$c(i,t) = \sqrt{\frac{2}{M}} \sum_{m=1}^{M} E(m,t)^\gamma \cos\left(\frac{\pi i}{M}(m-0.5) \right), \quad i = 1,2,...,l \tag{3.24}$$

where γ is the root value used to compress the filter bank energies, $0 < \gamma \leq 1$, and M is the number of filters.

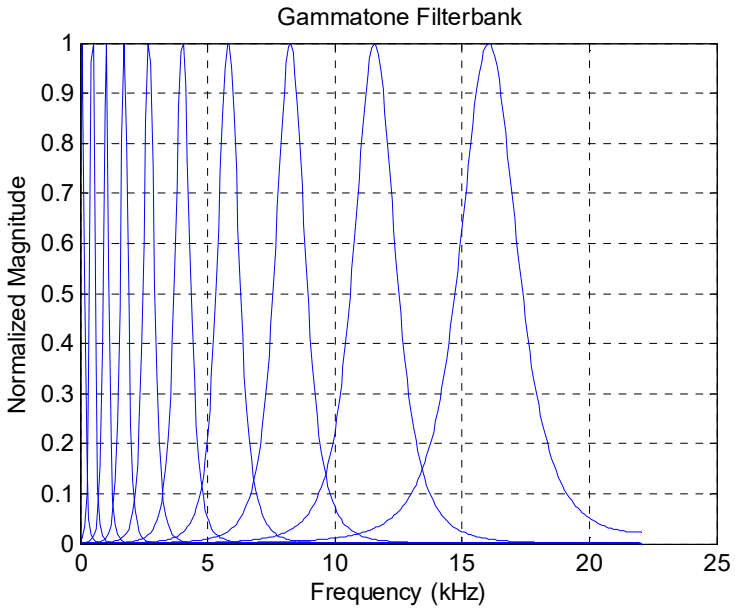


Figure 3.5: Example of a 10 channel gammatone filter bank

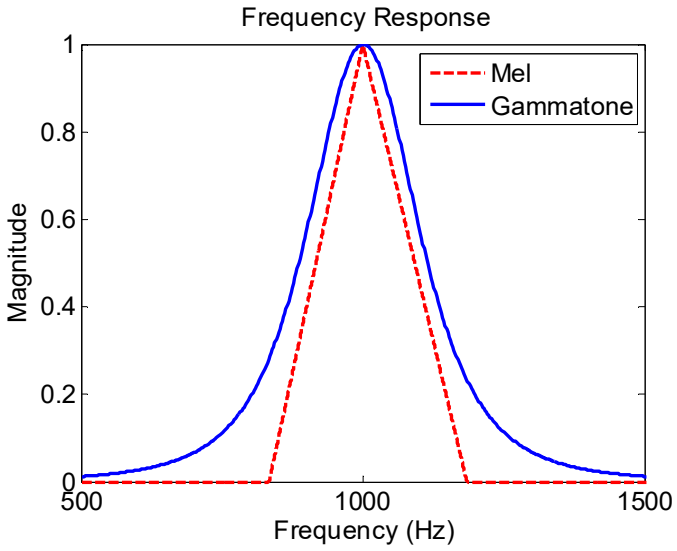


Figure 3.6: Frequency response of mel and gammatone filters at a center frequency of approximately 1 kHz

Cepstral Scaling

The feature vector for each sound file is generally represented by the mean and standard deviation along each feature dimension. However, to reduce the effect of different environmental conditions, the coefficients are often normalized before

feature vector formation. The data was normalized by scaling it in the range [0 1], referred as cepstral scaling (CS), which can be given as

$$\hat{c}(i,t) = \frac{c(i,t) - \min\left(c(i)\right)}{\max\left(c(i)\right) - \min\left(c(i)\right)} \tag{3.25}$$

where $max\left(c(i)\right)$ and $min\left(c(i)\right)$ are the maximum and minimum data values along the i^{th} feature dimension, respectively. The same formula also applies to delta and delta-delta coefficients.

Cepstral mean and variance normalization (CMVN) was also considered but the results using CS were generally found to be better.

3.1.4 Spectrogram Image Feature

The procedure for time-frequency image generation and feature extraction is explained with reference to Figure 3.7.

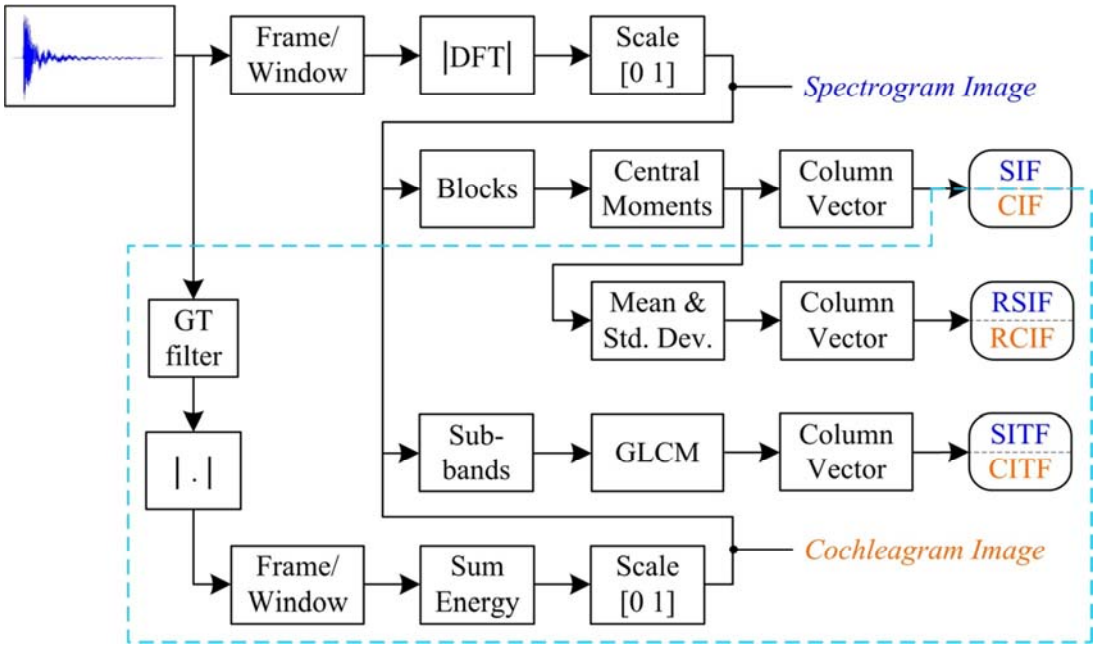


Figure 3.7: Steps in time-frequency image generation and feature extraction (proposed methods are enclosed in dashed lines).

For the baseline time-frequency image feature, that is, the SIF, central moments are extracted as features from the spectrogram images. Linear and log spectrogram images are considered in this work. To obtain the spectrogram images, the linear and log spectrum values are firstly obtained from the DFT values as

$$S_{linear}(k,t) = \left|X(k,t)\right| \tag{3.26}$$

and

$$S_{log}(k,t) = \log\left|X(k,t)\right|, \tag{3.27}$$

respectively.

These values are then normalized in the range [0,1] which gives the grayscale spectrogram image intensity values. The normalization is given as

$$I(k,t) = \frac{S(k,t) - \min(S)}{\max(S) - \min(S)}. \tag{3.28}$$

Illustrations of linear and log spectrogram images under clean conditions and with the addition of noise at 0dB SNR can be found in Figure 3.8 for a sample sound signal from *construction* sound class. Color representations are shown for the grayscale values for better visualization.

Each time-frequency image is divided into blocks and the v^{th} central moment for any given block of image is then determined as

$$\mu_v = \frac{1}{K}\sum_{i=1}^{K}(I_i - \mu)^v \tag{3.29}$$

where K is the sample size or the number of pixels in the block, I_i is the intensity value of the i^{th} sample in the block, and μ is the mean intensity value of the block.

3.2 Proposed Methods

This section presents the proposed features, RSIF and SITF, and the proposed time-frequency image representation, cochleagram. The steps in the proposed feature extraction and time-frequency image generation are given in Figure 3.7.

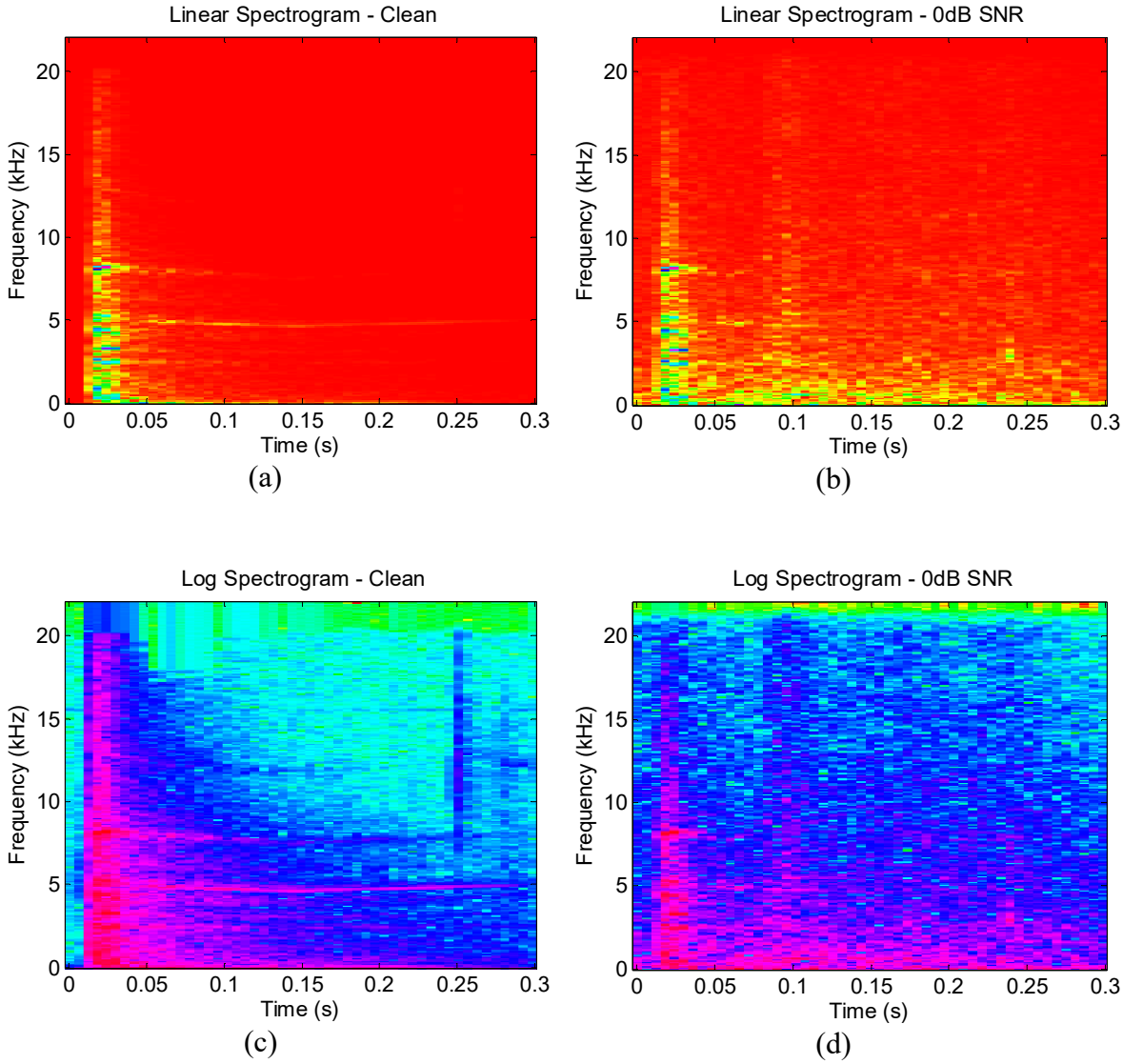


Figure 3.8: Linear and log spectrogram images of a sound signal from *construction* sound class. (a) Linear spectrogram image under clean conditions, (b) linear spectrogram image at 0dB SNR with factory noise, (c) log spectrogram image under clean conditions, and (d) log spectrogram image at 0dB SNR with factory noise.

3.2.1 Reduced Spectrogram Image Feature

The feature vector representation for the SIF has a drawback. The final feature vector is a concatenation of the central moment values computed in each block. However, if the sound signal segmentation is not similar, which will be especially true for non-stationary signals, the location of the same block in the spectrogram image of two sound signals of the same class may refer to different spectral regions, making the classification task difficult.

In addition, if the number of blocks along the rows and columns of the spectrogram image are same and is given as B, the dimension of the final feature vector using this approach is B^2. While this gives a reasonable size feature dimension for small values of B, the feature vector dimension can become extremely large as the number of blocks increases. In [2], the spectrogram images are divided into 9×9 blocks. The final feature dimension with two features, second and third central moments, computed in each block is $9 \times 9 \times 2 = 162$.

This study proposes an alternative feature data representation technique that to some extent negates the effect of inconsistent segmentation and also significantly reduces the feature vector dimension, therefore, referred as the reduced SIF (RSIF). The procedure is same as the SIF but the mean and standard deviation of the central moment values along the rows and columns of the image blocks are concatenated to form the feature vector as depicted in Figure 3.9. As such, the central moment values in each block will be same as the SIF but using statistical representation of feature data means inconsistencies in segmentation will be evened out. Also, the RSIF gives a feature vector dimension of $B \times 4$. While the feature dimension is higher than the SIF for $B < 4$, it gives a lower feature dimension for $B > 4$. Using the case of 9×9 blocks once again, the final feature dimension is $9 \times 4 \times 2 = 72$ which is 2.25 times smaller than the SIF. However, the preference of one feature data representation method over another is largely dependent on the classification performance which is compared in Chapter 5.

3.2.2 Spectrogram Image Texture Feature

The intensity values in a spectrogram image are determined by the spectral energy in the sound signal at any given time and frequency. The dominant frequency components in the sound signal are mostly unaffected by the noise as long as the noise signal does not contain strong spectral peaks, as shown in the linear spectrogram image in Figure 3.8(b) with *factory* noise. As such, the proposed SITF aims to capture the patterns of the subband spectral energy in trying to achieve noise robust classification performance.

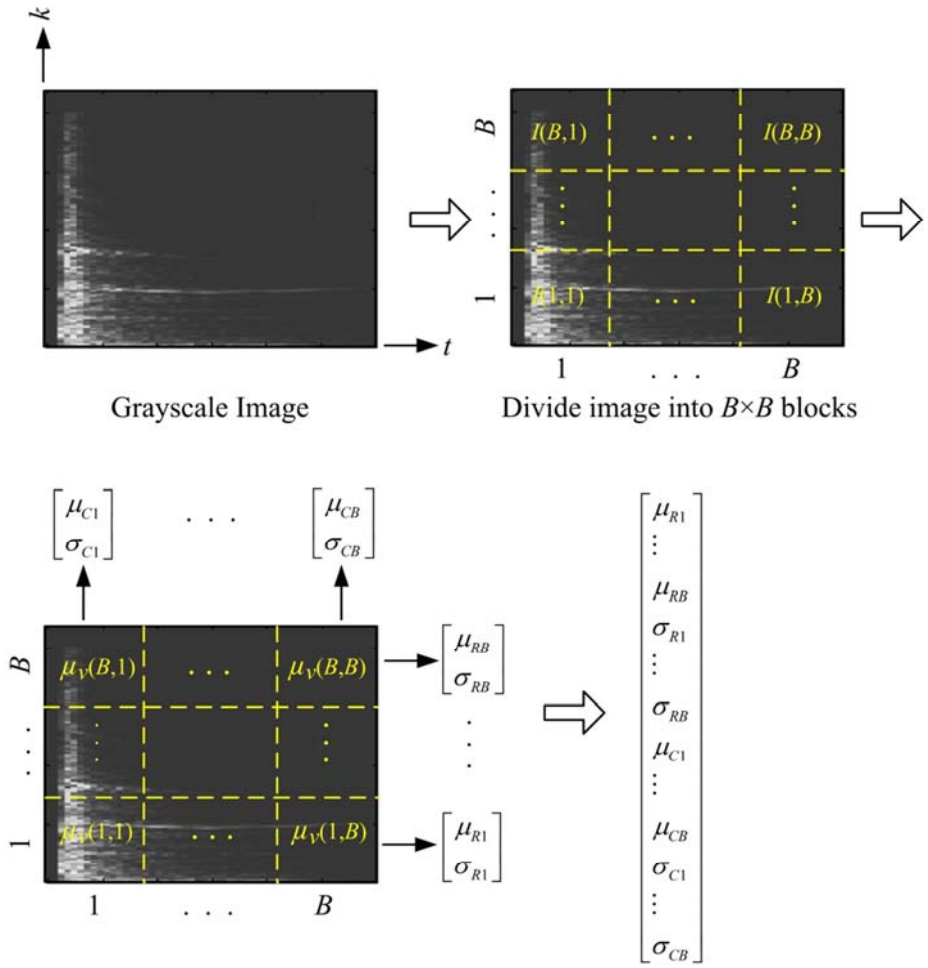


Figure 3.9: RSIF data representation. Note that $I(b,b)$ is a matrix of image intensity values for the block in the b^{th} row and b^{th} column, $\mu_v(b,b)$ is the v^{th} central moment for the block in the b^{th} row and b^{th} column, and μ_{Rb}, σ_{Rb} and μ_{Cb}, σ_{Cb} are the mean and standard deviation of the extracted feature for the blocks in the b^{th} row and b^{th} column, respectively, $b = 1, 2, \dots, B$.

The SITF uses the GLCM method of texture analysis which is a matrix of frequencies where each element (i,j) is the number of times intensity value j is located at a certain distance and angle, given by the displacement vector $[d_k\ d_t]$, where d_k is the offset in the y direction and d_t is the offset in the x direction, from intensity value i in an $N_t \times N_k$ image I. Mathematically, this can be given as

$$P(i,j)=\sum_{k=1}^{N_k}\sum_{t=1}^{N_t}\begin{cases}1, \text{ if } I(k,t)=i \ \& \ I(k+d_k,t+d_t)=j\\ 0, \text{ otherwise}\end{cases} \tag{3.30}$$

where the size of the output matrix is $N_g \times N_g$, N_g is the number of quantized gray levels. The typical angles for computing the GLCM are $0°, 45°, 90°, \text{and } 135°$ corresponding to the displacement vector $[0\ d]$, $[-d\ d]$, $[-d\ 0]$, and $[-d\ -d]$, respectively, as depicted in Figure 3.10. The feature vector for SITF is then formed by concatenating the GLCM values into a column vector.

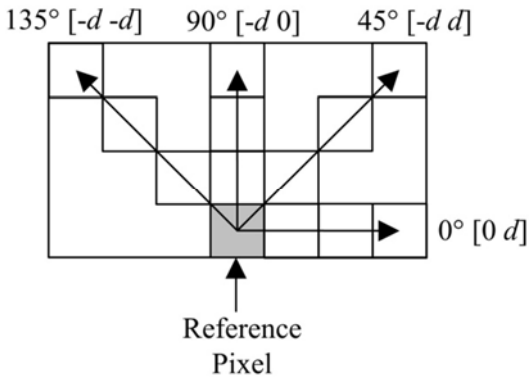


Figure 3.10: Directionality used in computing GLCM

3.2.3 Cochleagram

The cochleagram is another form of time-frequency representation and is based on the components of the outer and middle ear [23]. In this representation, the signal is broken into different frequencies which are naturally selected by the cochlea and hair cells. This frequency selectivity can be modeled by a filter bank, such as a gammatone filter.

A representation similar to the conventional spectrogram image can be obtained by smoothing the time series associated with each frequency channel of the gammatone filter and then adding the energy in the windowed signal for each frequency component which can be given as

$$C(m,t)=\sum_{n=0}^{N-1}\left|\hat{x}(m,n)\right|w(n), \qquad m=1,2,...,M_2 \tag{3.31}$$

where $\hat{x}(n)$ is the gammatone filtered signal and $C(m,t)$ is the m^{th} harmonic corresponding to the center frequency f_{cm} for the t^{th} frame.

These values are then normalized using (3.28) to get the grayscale cochleagram image intensity values. Illustrations of linear cochleagram images under clean conditions and with the addition of noise at 0dB SNR are given in Figure 3.11(a) and (b), respectively, using the same sound signal as the spectrogram images of Figure 3.8.

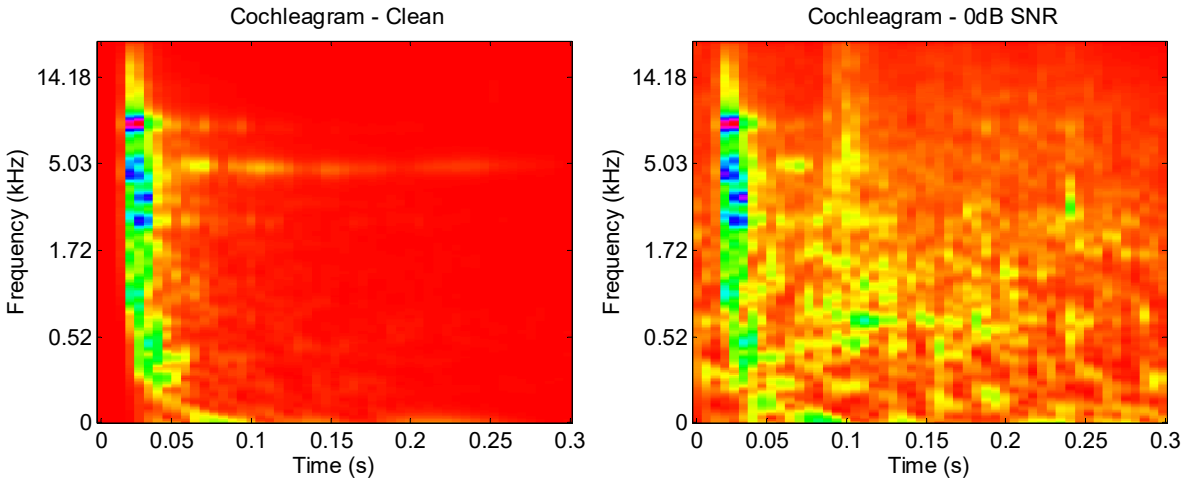


Figure 3.11: Linear cochleagram images for a sample sound signal from *construction* sound class. (a) Linear cochleagram image under clean conditions and (b) linear cochleagram image at 0dB SNR with *factory* noise.

3.2.4 Motivation

The GLCM essentially captures the frequency of repeating patterns or intensity value combinations in the time-frequency image. This work uses only two intensity levels, $N_g = 2$, as determined to give the best results in [17]. This means that the grayscale time-frequency image is essentially treated as a binary image for feature extraction, therefore, revealing only the dominant frequency components. This also means that small linear transformations caused by the noise to the intensity values of the sound signal in the time-frequency image would not affect its transformation to binary format as long as the threshold for binary conversion is not crossed. In addition, as shown in the linear time-frequency image in Figure 3.8(b) and Figure

3.11(b), the noise significantly affects only certain frequency bands and the use of subband feature extraction, with the optimal number of subbands determined as 64 in [17], ensures that feature data in subbands not seriously affected by noise largely remain unchanged.

This is better illustrated in Figure 3.12(a) and (b) where the normalized spectral energy distributions of a sound signal for the spectrogram and cochleagram images are shown, respectively. The spectral energy, in this context measured as the number of white pixels in the binary transformed image, is given in each of the 64 subbands without noise and with noise at 0dB SNR. The noise mostly affects subbands 13, 18, and 19 in the spectrogram image and subbands 40, 45, and 46 in the cochleagram image. Otherwise, there is generally a good degree of correlation between the energy distributions of the clean and noisy signals in both representations. As such, except in these bands, the repeating patterns captured by the GLCM will largely remain unchanged from clean to 0dB SNR conditions, explaining the usefulness of the proposed feature extraction technique.

In addition, while the spectrogram and cochleagram images of Figure 3.8 and Figure 3.11, respectively, use the same frequency range, $\left[0, {}^{F_s}/_{2}\right]$, the cochleagram offers a number of advantages [23]. Firstly, with the ERB spacing of the filter center frequencies, the cochleagram offers an expanded representation at low frequencies, where most of the spectral information lies for the sound signals used in this work. Secondly, depending on the type of sound signal, formants in the lower frequencies can be resolved into harmonics in the cochleagram since they have a narrower bandwidth. Therefore, a cochleagram offers more frequency components in the lower frequency range with narrower bandwidth and fewer frequency components in the higher frequency range with wider bandwidth, showing more spectral information than a spectrogram, as a result. The cochleagram also emphasizes acoustic onsets which can be effective for audio separation [86].

The difference in the spread of spectral energy for the two representations is also illustrated in Figure 3.12. For example, for the spectrogram image, the spectral energy is mainly distributed between subbands 2 to 20 and subbands 26 to 59 for the cochleagram image, that is, over 18 subbands for the spectrogram image and 33 subbands for the cochleagram image. As such, the cochleagram image clearly

reveals more spectral information which makes it a more effective time-frequency representation for feature extraction.

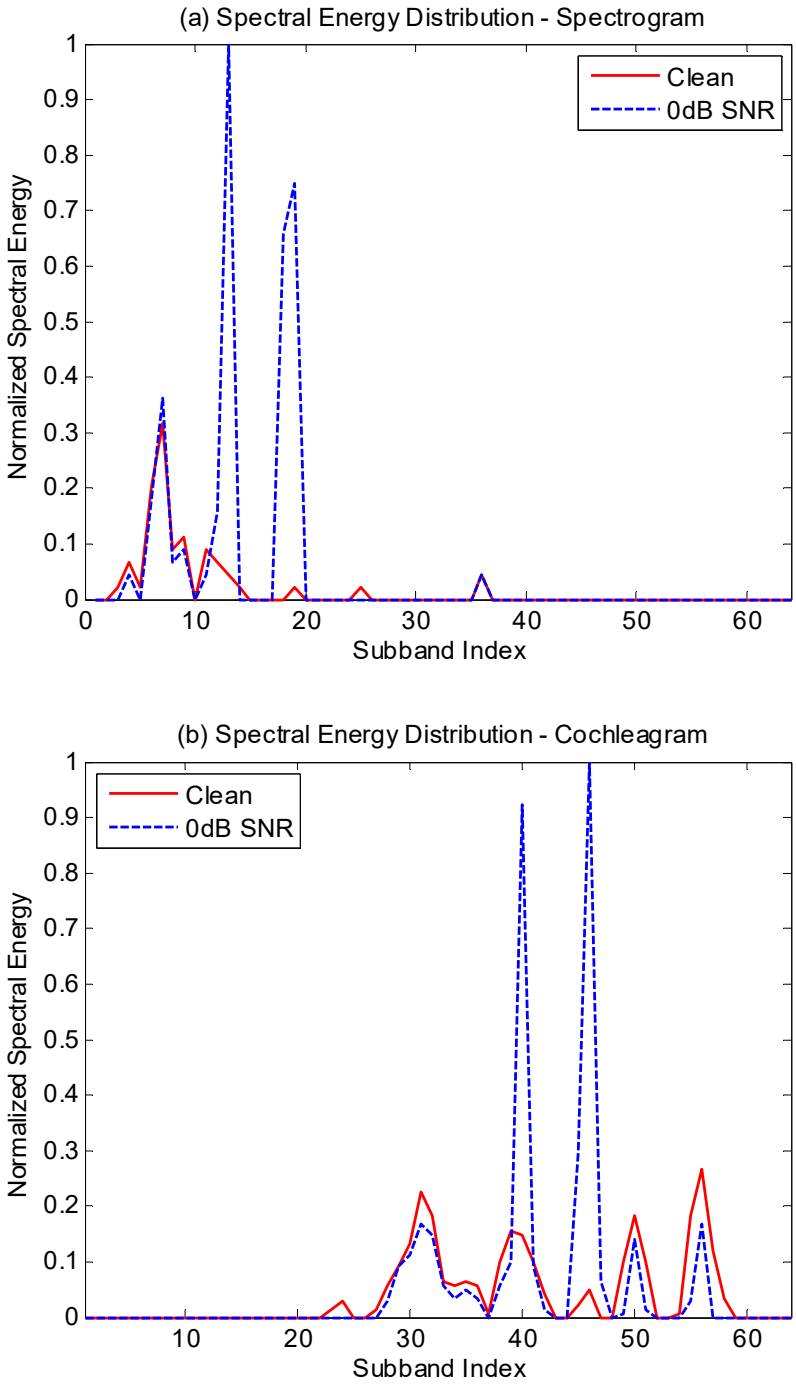


Figure 3.12: Subband spectral energy distribution of a sound signal from *construction* sound class with and without noise for (a) spectrogram and (b) cochleagram.

Chapter 4

Classification Methods

4.1 Baseline Methods

Various classification methods have been proposed for the various pattern recognition problems over past decades. In this work, two such methods are chosen as baseline methods which are *k*NN and SVM. *k*NN is one the earliest and the simplest of all machine learning algorithms while SVM is a relatively new classifier which has been shown to be on par, and in some cases better than, the more traditional classification methods such as HMM and GMM.

4.1.1 *k*-Nearest Neighbor

In *k*NN classification, the unknown test sample is classified to the majority vote of its neighbors from all the training samples. The Euclidean distance, the most commonly used distance measure and which has been used in this work, between two feature vectors **p** and **q** is the length of the line segment connecting them and can be given as

$$\overline{\mathbf{pq}} = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \ldots + (q_d - p_d)^2} = \sqrt{\sum_{i=1}^{d} (p_i - q_i)^2} \tag{4.1}$$

where $\mathbf{p} = (p_1, p_2, \ldots, p_d)$, $\mathbf{q} = (q_1, q_2, \ldots, q_d)$, and d is the feature vector dimension.

4.1.2 Support Vector Machines

SVM - Basic Theory

SVM has been well described in many literature, such as in [88, 89, 157], and is summarized here. SVM determines the optimal hyperplane to maximize the distance between any two given classes. Starting with a case of linearly separable dataset, consider a set of l training samples belonging to two classes, a positive class and a negative class, given as $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$, where $\mathbf{x}_i \in R^d$ is a d-dimensional feature vector representing the i^{th} training sample, and $y_i \in \{-1, +1\}$ is the class label of $\mathbf{x}_i$. There can be many possible hyperplanes but the two classes can be said to be optimally separated by the hyperplane if the separation distance, or margin, between the closest vector, known as support vectors, to the hyperplane is maximal, as shown for the two-dimensional linearly separable problem in Figure 4.1.

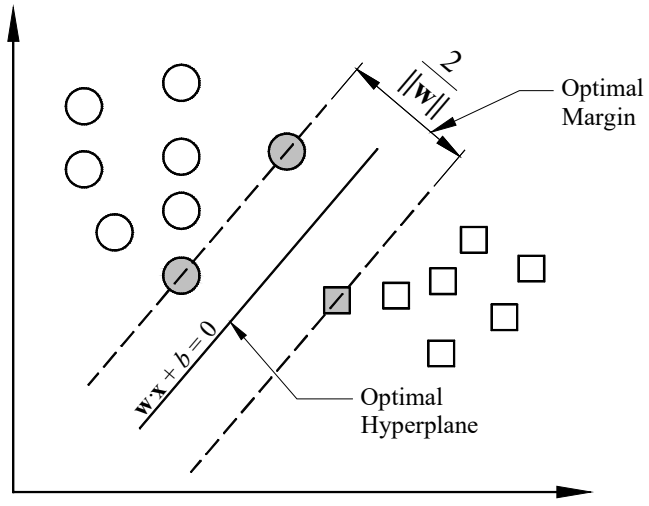


Figure 4.1: An example of a two-class linearly separable problem with the largest margin given by the lines passing through the support vectors (shaded in gray).

Any hyperplane in the feature space can be described by the equation

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{4.2}$$

where $\mathbf{w} \in R^d$ is a normal vector to the hyperplane and b is a constant. Selecting two hyperplanes,

$$\mathbf{w} \cdot \mathbf{x} + b = +1 \tag{4.3}$$

and

$$\mathbf{w} \cdot \mathbf{x} + b = -1 \tag{4.4}$$

such that the data points are separated with no data between them in the margin region, the aim then is to maximize the distance between them. The distance between these two hyperplanes is given as $^{\mathbf{2}}/_{\|\mathbf{w}\|}$, therefore, $\|\mathbf{w}\|$ has to be minimized. To prevent the data points from falling into the margin, the following constraints are added:

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq +1 \text{ for } y_i = +1 \tag{4.5}$$

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \text{ for } y_i = -1. \tag{4.6}$$

This can be rewritten in the equivalent form as

$$y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b \right) \geq 1, \ i = 1, ..., l. \tag{4.7}$$

The optimization problem can then be stated as

$$\arg \min_{(\mathbf{w},b)} \left(\|\mathbf{w}\| \right) \tag{4.8}$$

$$\text{subject to: } y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b \right) \geq 1, \ i = 1, ..., l. \tag{4.9}$$

For mathematical convenience, and without altering the solution, $\|\mathbf{w}\|$ is substituted with $½\|\mathbf{w}\|^2$ and this quadratic programming problem can now be given as

$$\arg \min_{(\mathbf{w},b)} \left(\frac{1}{2} \|\mathbf{w}\|^2 \right) \tag{4.10}$$

$$\text{subject to: } y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b \right) \geq 1, \ i = 1, ..., l. \tag{4.11}$$

The optimization problem can be solved under the given constraints by the saddle point of the Lagrange functional

$$\arg \min_{(\mathbf{w},b)} \max_{(\alpha)} \left(\frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i \left(y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b \right) - 1 \right) \right) \tag{4.12}$$

with the Lagrange multipliers $\alpha_i \geq 0$. For ease of computation, this primal problem is transformed to a dual problem using classical Lagrangian duality which reduces to the following optimization problem

$$\arg\max_{(\alpha)}\left(\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j\left(\mathbf{x}_i\cdot\mathbf{x}_j\right)\right) \tag{4.13}$$

$$\text{subject to: } \alpha_i\geq 0,\ i=1,...,l \text{ and } \sum_{i=1}^{l}\alpha_i y_i=0. \tag{4.14}$$

This gives the solution

$$\mathbf{w}=\sum_{i=1}^{l}\alpha_i y_i\mathbf{x}_i. \tag{4.15}$$

The $\mathbf{x}_i$ for which $\alpha_i>0$ are called the supported vectors which lie exactly on the margin satisfying $y_i(\mathbf{w}\cdot\mathbf{x}_i+b)=1$. The remaining data samples are irrelevant since their multipliers satisfy $\alpha_i=0$.

The offset can then be determined as

$$b=y_i-\mathbf{w}\cdot\mathbf{x}_i \tag{4.16}$$

using any support vector or averaged over all support vectors.

However, there is no such hyperplane for linearly nonseparable problems to classify every training sample correctly. In such a case, the optimization can be generalized by introducing the concept of *soft margin* implying a hyperplane separating most but not all the points. Introducing non-negative *slack* variables ξ_i which measure the degree of misclassification of data $\mathbf{x}_i$ and a penalty function $\sum_i\xi_i$, the optimization is a trade-off between a large margin and a small error penalty and can be given as

$$\arg\min_{(\mathbf{w},b,\xi)}\left(\frac{1}{2}\|\mathbf{w}\|^2+T\sum_{i=1}^{l}\xi_i\right) \tag{4.17}$$

$$\text{subject to: } y_i\left(\mathbf{w}\cdot\mathbf{x}_i+b\right)\geq 1-\xi_i,\ i=1,...,l \text{ and } \xi_i\geq 0 \tag{4.18}$$

where T is a penalty or tuning parameter to balance the margin and training error.

This optimization problem can be solved under the given constraints by the saddle point of the Lagrangian given as

$$\arg\min_{(\mathbf{w},b,\xi)}\max_{(\alpha,\beta)}\left(\frac{1}{2}\|\mathbf{w}\|^2+T\sum_{i=1}^{l}\xi_i-\sum_{i=1}^{l}\alpha_i\left(y_i\left(\mathbf{w}\cdot\mathbf{x}_i+b\right)-1+\xi_i\right)-\sum_{i=1}^{l}\beta_i\xi_i\right) \tag{4.19}$$

with Lagrange multipliers $\alpha_i \geq 0$ and $\beta_i \geq 0$. As before, the primal problem is transformed to a dual problem using classical Lagrangian duality as

$$\arg\max_{(\alpha)} \left(\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j \right) \right) \tag{4.20}$$

$$\text{subject to: } 0 \leq \alpha_i \leq T,\ i = 1,...,l \text{ and } \sum_{i=1}^{l} \alpha_i y_i = 0. \tag{4.21}$$

Due to the linear penalty function, the slack variables do not appear in the dual formulation of the problem and the solution is same as the separable case except for a modification to the Lagrange multipliers: $0 \leq \alpha_i \leq T, i = 1, \dots, l$.

In applications where linear SVM does not give satisfactory results, nonlinear SVM is suggested which aims to map the input vector $\mathbf{x}$ to a higher dimensional space $\mathbf{z}$ through some nonlinear mapping $\phi(\mathbf{x})$ chosen *a priori* to construct an optimal hyperplane. The *kernel trick* [88] is applied to create the nonlinear classifier where the dot product is replaced by a nonlinear kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ which computes the inner product of the vectors $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$.

The typical kernel functions are:

- polynomial, $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^r$ where r is the degree of the polynomial;
- Gaussian RBF, $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2\right)$, where $\sigma > 0$ is the width of the Gaussian function; and
- multilayer perception, $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(a_1(\mathbf{x}_i \cdot \mathbf{x}_j) - a_2)$, where a_1 and a_2 are two given parameters known as *scale* and *offset* respectively.

The classifier for a given kernel function with the optimal separating hyperplane is then given as

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \right). \tag{4.22}$$

Multiclass Classification

While many multiclass SVM classification methods have been proposed over the years, four commonly used methods, OAA, OAO, DDAG, and ADAG, are considered in this work. OAA, which is probably the earliest of the multiclass SVM classification techniques [89], distinguishes between one of the class labels against the rest. It uses a winner-takes-all strategy in which the classifier that has the highest output function assigns the class. The OAO approach distinguishes between every pair of classes and classification is done using a max-wins voting strategy [90]. Every classifier assigns the instance to one of the two classes with the vote for the assigned class increased by one. In the end, the class with the most votes assigns the class label. DDAG [91] and ADAG [92] are also based on classification between pair of classes but utilize a decision tree structure in the testing phase.

One-Against-All SVM

Consider an M-class problem with l training samples: $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$, where $\mathbf{x}_i \in R^d$ is a d-dimensional feature vector representing the i^{th} training sample, and $y_i \in \{1,2, \ldots, M\}$ is the class label of $\mathbf{x}_i$. In the OAA approach, M binary SVM classifiers are constructed and evaluated where each classifier separates one class from all the other classes combined. That is, the i^{th} classifier is trained with all the training samples from the i^{th} class as positive labels and all the remaining samples as negatives labels.

The i^{th} SVM solves the following optimization problem:

$$\arg \min_{\left(\mathbf{w}^i, b^i, \xi^i\right)} \left(\frac{1}{2} \left\| \mathbf{w}^i \right\|^2 + T \sum_{i=1}^{l} \xi_j^i \right) \tag{4.23}$$

$$\text{subject to: } \tilde{y}_j \left(\mathbf{w}^i \cdot \phi\left(\mathbf{x}_j\right) + b^i \right) \geq 1 - \xi_j^i, \ \xi_j^i \geq 0 \tag{4.24}$$

where $\tilde{y}_j = 1$ if $y_j = i$ and $\tilde{y}_j = -1$ otherwise.

During classification, a sample $\mathbf{x}$ is classified in the class with the largest value of the decision function

$$f\left(\mathbf{x}\right) = \arg \max_{i=1,2,\ldots,M} \left(\mathbf{w}^i \cdot \phi\left(\mathbf{x}\right) + b^i \right). \tag{4.25}$$

The disadvantage of OAA method is the high mismatch in the training samples between the positive and negative classes while some literature [91, 158] also shows that the training and evaluation times are relatively high.

One-Against-One SVM

For an M-class problem, OAO method constructs and evaluates $M(M-1)/2$ classifiers where each SVM is trained on samples from two classes at a time. For the training samples from the i^{th} and j^{th} class, the following binary classification problem needs to be solved:

$$\arg \min_{\left(\mathbf{w}^{ij}, b^{ij}, \xi^{ij}\right)} \left(\frac{1}{2} \left\| \mathbf{w}^{ij} \right\|^2 + T \sum_{t=1}^{l} \xi_t^{ij} \right) \tag{4.26}$$

$$\text{subject to: } \tilde{y}_t \left(\mathbf{w}^{ij} \cdot \phi\left(\mathbf{x}_t\right) + b^{ij} \right) \geq 1 - \xi_t^{ij}, \ \xi_t^{ij} \geq 0 \tag{4.27}$$

where $\tilde{y}_t = 1$ if $y_t = i$ and $\tilde{y}_t = -1$ otherwise.

During classification, the class label of a test sample can be predicted as

$$f\left(\mathbf{x}\right) = \arg \max_{i=1,2,\ldots,M} \sum_{j=1, j\neq i}^{M} \operatorname{sgn}\left(\mathbf{w}^{ij} \cdot \phi\left(\mathbf{x}\right) + b^{ij} \right). \tag{4.28}$$

While the OAO method has much more uniform training samples in the positive and negative classes when compared to OAA method, its disadvantage is the inefficiency of classifying data because the number of SVM classifiers grows super linearly with an increase in the number of classes. DDAG and ADAG techniques remedy this disadvantage using a decision tree architecture.

Decision Directed Acyclic Graph SVM

DAG is a graph where the edges have an orientation and no cycle. The structure of a rooted binary DAG by Platt et al. [91] is shown in Figure 4.2. A rooted binary tree has nodes arranged in a triangle. The single root node is at the top, two nodes in the second layer, and so on with M leaves in the last layer where M is the number of classes. The i^{th} node in layer $j < M$ is connected to the i^{th} and $(i+1)^{th}$ node in the $(j+1)^{th}$ layer.

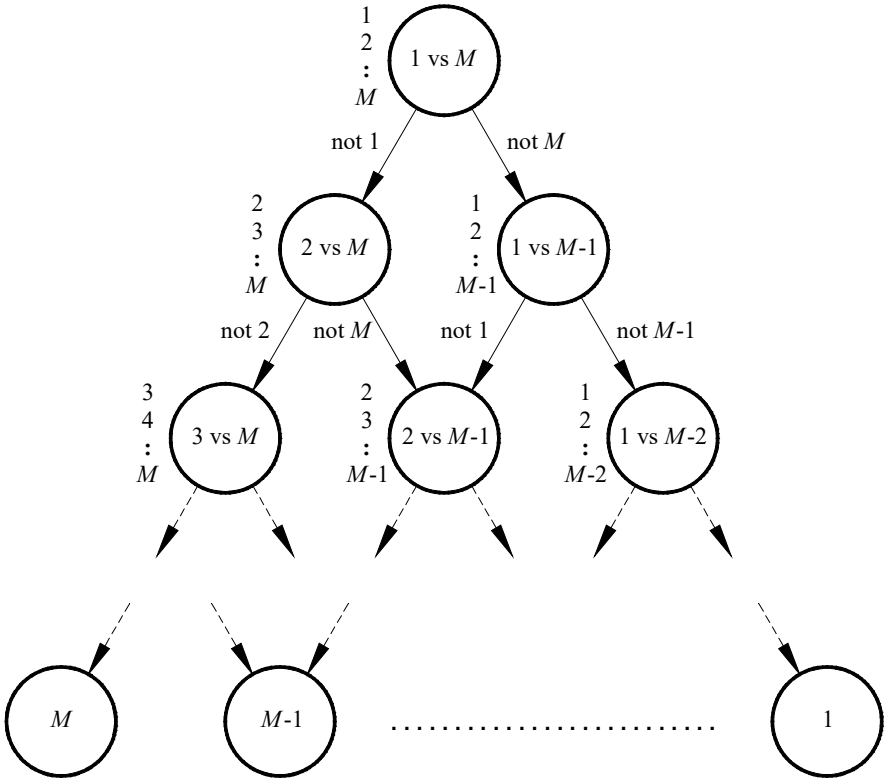


Figure 4.2: DDAG structure for an M-class problem. The root node is at the top of the tree and there are M-leaves at the bottom of the tree. Evaluation starts at the root node from where each class is removed from the class order list at each node. Only one class is left at the leaf node which is the decision function.

The evaluation of a DDAG starts at the root node and, depending of the outcome of the binary function, the node is exited through the left edge if the outcome is zero and the right edge otherwise. The binary function at the next node is then evaluated and this continues until the leaf node is reached, which is the value of the decision function. The DDAG operates on a class order list which is initialized at the root node. The list is updated at each subsequent node where one class is eliminated from the list. The evaluation at each node corresponds to the first and last classes in the list. There is only one class left in the list after $M - 1$ evaluations. At this point, the leaf node has been reached and the path taken from the root node to the leaf is called the evaluation path. As mentioned in [91], the choice of the class order in the list is arbitrary and in their experimentation, a class list in numerical/alphabetical order was used since a few different combinations of class order did not show significant changes in the accuracy.

Similar to the OAO method, the DDAG method creates $M(M-1)/2$ nodes during training phase but only $M-1$ nodes are evaluated during testing. As such, DDAG outperforms OAO in terms of computation speed. However, as pointed out by Kijsirikul et al. in [92], the node evaluations for the correct class is unnecessarily high which creates high cumulative error. On average, the number of times a correct class has to be tested against other classes scales linearly with M. In a worst case scenario, if the correct class is evaluated at the root node, it will be tested $M-1$ times, that is, tested against all the other classes, before being correctly classified.

Adaptive Directed Acyclic Graph SVM

Adaptive DAG is proposed by Kijsirikul et al. in [92] aimed at overcoming the shortcomings of DDAG method. Similar to DDAG, for an M-class problem, $M(M-1)/2$ binary classifiers are trained and $M-1$ evaluations are required during testing. However, an ADAG has a reversed triangular structure when compared to a DDAG as shown in Figure 4.3 for an M-class problem where M is assumed to be an even number for now.

Similar to DDAG, ADAG is implemented using a class order list, each node evaluates two classes, and a class is eliminated at each node. The classification starts at the top layer and based on the outcome of the binary function, the outgoing edge from the node passes the preferred class information to the next node. The top layer has $M/2$ nodes, the second layer has $M/2^2$ nodes, and so on. In general, the number of nodes in each layer is equal to $M/2^p$ where $p = 1, 2, \dots, P$ is the layer number starting from the top layer.

The elimination process continues at each node with the number of classes reducing by half in each layer until the final node, the output of which is the decision function. While the same number of evaluations are required as in DDAG-SVM, the number of evaluations that the correct class has to go through is $\lceil \log_2 M \rceil$, which is also equal to the number of layers, when compared to a maximum of $M-1$ evaluations for the correct class in DDAG method. In the case of odd number of classes, the last class in the list is not evaluated at a node until the number of classes in the list becomes even.

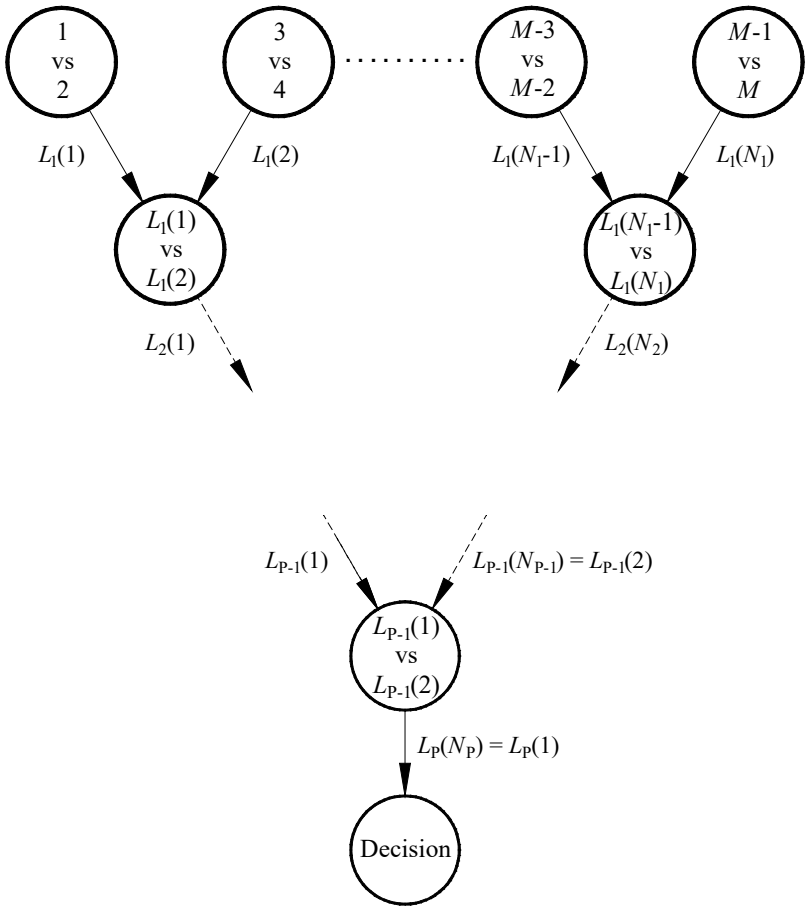


Figure 4.3: ADAG structure for an M-class problem (M assumed to be even) where L_p is the p^{th} layer, N_p is the number of nodes in the p^{th} layer, $L_p(q)$ is the output of the q^{th} node in the p^{th} layer, $q = 1, 2, \dots, N_p$, and $p = 1, 2, \dots, P$; $p = 1$ is the top layer.

4.2 Deep Neural Networks

Advancements in machine learning algorithms can significantly improve the classification performance in pattern recognition problems. As mentioned in [105], one such advancement in ASR was the introduction of expectation maximization (EM) algorithm [159] for representing the relationship between the HMM states and the acoustic input using GMMs. Such techniques were also employed in SER applications as in [2]. While ANNs trained using back propagating error derivatives also had the potential to learn more accurate models, limitations in hardware and learning algorithms for training neural networks with many hidden layers and large amounts of data restricted progress along these lines. However, this changed over

the last few years with advancements in computer hardware and machine learning algorithms giving rise to a modified machine learning algorithm called DNNs which has been shown to outperform GMMs for acoustics modeling in ASR on many different datasets by a number of research groups as summarized in [105]. The methods for DNNs is now available in a number of literature, such as [105, 106, 160], and is summarized here.

A DNN, as defined in [105], is a feed-forward ANN with more than one hidden layers of units between the input and output layers. The training data in a DNN can be modeled using a two-layer network known as a restricted Boltzmann machine (RBM). RBMs were invented by Smolensky in 1986 [161] but only gained attention in early 2000s after development of fast learning algorithms by Hinton [162]. A RBM is a generative energy based model that consists of a layer of stochastic binary visible units with undirected connections to a layer of binary hidden units, as shown in Figure 4.4, but no visible-visible or hidden-hidden connections.

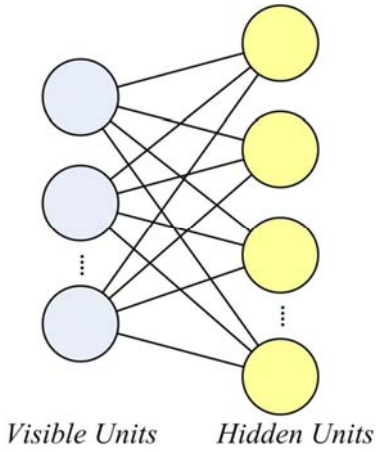


Figure 4.4: A restricted Boltzmann machine with visible and hidden layer connections

The DNN classifier [160] has L-layers with the feature vectors on the input layer and the output layer in a one-of-M configuration (M-classes). The DNN is constructed using individual pre-trained RBM pairs with each pair comprising V visible and H hidden stochastic nodes, $\mathbf{v} = [v_1, v_2, \dots, v_V]^T$ and $\mathbf{h} = [h_1, h_2, \dots, h_H]^T$. This work uses Bernoulli-Bernoulli RBM (BBRBM) structures for all layers, however, the

input layer can also be formed using Gaussian-Bernoulli RBM (GBRBM) structures as in [106]. Assuming binary nodes for the BBRBM structure, that is, $\mathbf{v}_{bb} \in \{0,1\}^V$ and $\mathbf{h}_{bb} \in \{0,1\}^H$, the energy function of the state $E_{bb}(\mathbf{v},\mathbf{h})$ can be given as

$$E_{bb}(\mathbf{v},\mathbf{h}) = -\sum_{i=1}^{V}\sum_{j=1}^{H} v_i h_j w_{ji} - \sum_{i=1}^{V} v_i b_i^v - \sum_{j=1}^{H} h_j b_j^h \tag{4.29}$$

where w_{ji} is the weight between the i^{th} visible unit and the j^{th} hidden unit and b_i^v and b_j^h are the real valued biases, respectively. The BBRBM model parameters are $\theta_{bb} = \{\mathbf{W}, \mathbf{b^h}, \mathbf{b^v}\}$ where the weight matrix is given as $\mathbf{W} = \{w_{ij}\}_{V\times H}$ with biases $\mathbf{b^h} = [b_1^h, b_2^h, \dots, b_H^h]^T$ and $\mathbf{b^v} = [b_1^v, b_2^v, \dots, b_V^v]^T$.

The joint probability associated with configuration $(\mathbf{v},\mathbf{h})$ can then be given as

$$p(\mathbf{v},\mathbf{h};\theta) = \frac{1}{Y} e^{\{-E(\mathbf{v},\mathbf{h};\theta)\}} \tag{4.30}$$

where Y is a partition function given as $Y = \sum_v \sum_h e^{\{-E(\mathbf{v},\mathbf{h};\theta)\}}$.

During pre-training, the training data is used to estimate the RBM model parameter θ with maximum likelihood learning using the contrastive divergence (CD) algorithm [104]. CD gives a simple approximation of the gradient of the log probability of the training data. A better generative model is learned through a limited number of steps of alternating Gibbs sampling by updating the hidden nodes $\mathbf{h}$ given the visible nodes $\mathbf{v}$ and then using the updated $\mathbf{h}$ to update $\mathbf{v}$. The training starts at the input layer, which is fed with the feature vectors, and form the visible nodes. The hidden units determined after the training process form the visible units for training the next RBM visible units. Multiple layers of RBMs are trained by repeating this process as many times as desired and, in the end, the RBMs are stacked to form a DNN as a single, multilayer generative model.

In fine-tuning, a *softmax* output labeling layer of size M is added which aims to convert a number of units in the final layer into a multinomial distribution using the *softmax* function

$$p(m \mid \mathbf{h_L};\theta_\mathbf{L}) = \frac{\phi(m,\theta_L)}{\sum_{p=1}^{M} \phi(p,\theta_L)} \tag{4.31}$$

where m is an index over all classes, θ_L are the model parameters for the DNN, $\phi(m,\theta_L) = e^{\{\sum_{i=1}^{H} w_{ki} h_i + b_m\}}$, and $p(m|\mathbf{h};\theta_L)$ is the probability of the input being classified into class m.

Back propagation derivatives of a cost function, which measures the discrepancy between the target outputs and the actual outputs for each training case [163], can then be used to discriminatively train the DNN. With the *softmax* output function, the cross entropy is the natural choice of cost function C between the desired and actual distributions given as

$$C = -\sum_{m=1}^{M} c_k \log p(m|\mathbf{h};\theta_L). \tag{4.32}$$

More on the setting for the various DNN parameters and the DNN structure for the various features considered in this work can be found in section 5.3.

Chapter 5

Experimental Evaluation

A description of the database of sounds used in this work is given first followed by an overview of the noise conditions and the experimental setup. Next, the results using the baseline features is presented, which include the two cepstral features, MFCCs and GTCCs, and the SIF. This is followed by the results for the proposed spectrogram image features, RSIF and SITF, and then the results for the three time-frequency image features using cochleagram feature extraction. Furthermore, results using feature vector combination are presented and the classification performance of the different classification methods is compared for all individual and combined features. Some further analysis is performed next which includes interclass classification performance, performance analysis of the different classification methods, and a comparison of the training and evaluation time of the various features.

5.1 Sound Database

The sound database has a total of 1143 files belonging to 10 classes. The choice of the sound classes is similar to other work in the area of audio surveillance such as [5, 7, 164]. The sound files are largely obtained from the RWCP Sound Scene database in Real Acoustic Environment [70] and the BBC Sound Effects library [54]. All signals in the database have 16-bit resolution and a sampling frequency of 44100 Hz. A summary of the selected sound classes, total number of sound files, and total duration is shown in Table 5.1.

Table 5.1: Overview of sound classes

	Class	Number of Subclasses	Total Number of Files	Total Duration (s)
Alarms	A	6	180	83.4533
Children Voices	B	6	180	131.9286
Construction	C	3	90	26.2251
Dog Barking	D	3	84	22.3042
Footsteps	E	6	171	24.0566
Glass Breaking	F	2	60	107.3296
Gunshots	G	3	84	8.9500
Horn	H	3	66	27.4115
Machines	I	3	90	56.8423
Phone Rings	J	6	138	119.7996
Total			**1143**	**608.3008**

Alarm sounds in the database include car alarms, electronic alarms, and siren. *Children voices* include children crying and screaming. *Construction* sounds are sawing, metal hammering, and pneumatic drilling. The *footstep* sounds include those from metal and wooden stairs and on pavement. The three types of *machine* sounds are machine hum, motor, and warble. The *phone rings* class includes cellphone and telephone ringtones.

The database has both harmonic and impulsive sounds and an irregular number of sound files which are important in testing out the robustness of the system. It is also important to have some degree of intraclass diversity and interclass similarity for this purpose and this is demonstrated using *k*-means clustering [165]. The centroid of each of the subclasses was determined and these were grouped into 10 clusters using *k*-means clustering algorithm. The results for these are shown in Table 5.2 where A_B and A_A show the subclasses in class A before and after applying *k*-means clustering, respectively.

As an example, there are six types of *alarm* sounds (class A_B) which have been labeled as $A_1, A_2, \ldots, A_6$. However, after applying *k*-means clustering, the six subclasses fall in five different clusters: A_1 and A_2 in class A_A, A_5 in class B_A, A_4 in class C_A, A_3 in class D_A, and A_6 in class H_A. This means that only A_1 and A_2 have similar signal properties. There are three subclasses in *construction*, class *C*, and all

fall in different clusters, B_A, C_A, and G_A, unlike the subclasses from *dog barking*, *glass breaking*, and *horn* which all fall in the same cluster, D_A, F_A, and H_A, respectively, but have been combined with subclasses from other classes.

Table 5.2: Demonstration of intraclass diversity and interclass similarity using *k*-means clustering

Normal Cluster							**After K-means Clustering**							
Class	**Subclasses**						**Class**	**Subclasses**						
$\mathbf{A_B}$	A_1	A_2	A_3	A_4	A_5	A_6	$\mathbf{A_A}$	A_1	A_2	J_2	J_3	J_4	J_6	
$\mathbf{B_B}$	B_1	B_2	B_3	B_4	B_5	B_6	$\mathbf{B_A}$	A_5	B_1	B_2	B_5	C_1	J_1	
$\mathbf{C_B}$	C_1	C_2	C_3				$\mathbf{C_A}$	A_4	C_2	I_1				
$\mathbf{D_B}$	D_1	D_2	D_3				$\mathbf{D_A}$	A_3	B_3	B_4	B_6	D_1	D_2	D_3
$\mathbf{E_B}$	E_1	E_2	E_3	E_4	E_5	E_6	$\mathbf{E_A}$	E_3	E_4	E_5	E_6			
$\mathbf{F_B}$	F_1	F_2					$\mathbf{F_A}$	F_1	F_2	G_3				
$\mathbf{G_B}$	G_1	G_2	G_3				$\mathbf{G_A}$	C_3	G_1	G_2				
$\mathbf{H_B}$	H_1	H_2	H_3				$\mathbf{H_A}$	A_6	H_1	H_2	H_3	J_5		
$\mathbf{I_B}$	I_1	I_2	I_3				$\mathbf{I_A}$	I_2						
$\mathbf{J_B}$	J_1	J_2	J_3	J_4	J_5	J_6	$\mathbf{J_A}$	E_1	E_2	I_3				

5.2 Noise Conditions

The performance of all features is evaluated under three different noise environments taken from the NOISEX-92 database [71]: *speech babble*, *factory floor 1*, and *destroyer control room*. As in [5], the signals are resampled at 44100 Hz and the performance is evaluated in clean conditions and at 20dB, 10dB, 5dB, and 0dB SNRs.

5.3 Experimental Setup

For all experiments, signal processing is carried out using a Hamming window of 512 points (11.61 ms) with 50% overlap. The classification accuracy is given in percentage as *number of correctly classified test samples* divided by the *total number of test samples*. Nonlinear SVM with a Gaussian RBF kernel is used in all

cases as it was found to give the best results. The penalty parameter T and σ for the Gaussian RBF kernel were tuned using cross validation. In tuning the parameters, one set of parameters which gave the best average classification accuracy were selected rather than determining the optimal parameters for each noise level. For DDAG and ADAG, the class order list in alphabetical order was used. The best k value for the kNN classifier was determined similarly in each experiment.

For the DNN classifier, the dimensions and number of hidden layers were determined through experimentation in each case, following a similar procedure to [106]. That is, a step-wise search of hidden layer widths between 10 and 400 was performed. The resolution in each case was set to 10 and the internal layers were constrained to equal size. Similar to [106], results are only presented using two hidden layers for all the features since the addition of more hidden layers was only seen to give a marginal improvement in classification performance but with significant increase in computation time. The final DNN structures for all features are given in Table 5.3 where the input and output layers are equal to the feature dimension and number of classes, respectively. In addition, for all experiments, the batch training size was set to 127, one-sixth of the number of training samples, and using 1000 training epochs.

Table 5.3: Final DNN structures for all feature vectors

Feature	DNN Structure			
	Input Layer	Internal Layer 1	Internal Layer 2	Output Layer
MFCCs and GTCCs	72	50	50	10
SIF and CIF	162	60	60	10
RSIF and RCIF	72	50	50	10
SITF and CITF	256	60	60	10
Linear GTCC + CIF	234	160	160	10
Linear GTCC + RCIF	144	100	100	10
Linear GTCC + CITF	328	160	160	10

For all experimentations, the classifier is trained with two-third of the clean samples with the remaining one-third samples used for testing under clean and noisy conditions. The OAA-SVM classification method is used as the baseline classifier for all experiments. This means that in trying to determine the optimal parameter settings for a feature, results are reported only using OAA-SVM classification method. This is because, in general, a good correlation was seen between all the classifiers considered in this work and hence, no need was seen to report results using all classifiers. However, at the optimal parameter settings for the features, classification accuracy using all or best classifiers is reported.

5.4 Results using Baseline Features

5.4.1 Log Cepstral Coefficients

The cepstral features, MFCCs and GTCCs, form the first set of baseline features. In computing the cepstral coefficients, log compression is applied to the filter bank energies for all experiments in this subsection.

Static, Delta, and Delta-Delta Coefficients

In the first experiment, the feature vector is formed using the *static* coefficients and then combined with the *delta* and *delta-delta* coefficients. With the *static* coefficients, the feature vector for each frame is 12-dimensional, 12 cepstral coefficients with the 0^{th} coefficient excluded. Similarly, for *static* + *delta*, each frame is 24-dimensional and 36-dimensional for *static* + *delta* + *delta-delta*. After data normalization, the final feature vector is represented by concatenating the mean and standard deviation for each feature dimension. This gives a 24-dimensional final feature vector for *static* coefficients, 48-dimensional for *static* + *delta*, and 72-dimensional for *static* + *delta* + *delta-delta*. In this initial experiment, for MFCCs and GTCCs, a 24-filter bank system is used with a frequency range of 0 to 22050 Hz, which is the Nyquist frequency.

The classification accuracy values for MFCCs and GTCCs averaged over clean samples and noisy samples, 20dB, 10dB, 5dB, and 0dB SNRs, are given in Table 5.4 using OAA-SVM classification. The results for GTCCs are given using all three ERB models. The inclusion of the first and second derivatives gives significant

improvement in the results for both MFCCs and GTCCs. For GTCCs, the average classification accuracy with Lyon's cochlear model is significantly better than the other two. It was noted that while all three models gave comparable classification accuracy under clean conditions, Lyon's cochlear model gave much better classification accuracy at 20dB, 10dB, 5dB, and 0dB SNRs which results in a better overall performance for GTCCs.

Table 5.4: Average classification accuracy values for MFCCs and GTCCs with different feature vector dimensions and different ERB models for GTCCs

	MFCC			GTCC								
				Glasberg & Moore [153]			Lyon [154]			Greenwood [155]		
Feature Dim.	24-D	48-D	72-D	24-D	48-D	72-D	24-D	48-D	72-D	24-D	48-D	72-D
Accuracy	60.75	69.68	**73.05**	57.03	64.90	67.12	62.80	71.67	**76.69**	56.76	62.94	65.67

For the best average classification accuracy for each cepstral feature, the classification accuracy values under each noise condition are given in Table 5.5. MFCCs and GTCCs give comparable classification accuracy under clean conditions and at 20dB SNR, however, the classification performance of GTCCs is seen to be significantly better at 10dB, 5dB, and 0dB SNRs.

Table 5.5: Classification accuracy values for the best average classification accuracy for MFCCs and GTCCs

Feature	Clean	20dB	10dB	5dB	0dB	Ave
MFCC	98.43	95.98	74.37	57.74	38.76	73.05
GTCC	98.69	95.10	78.22	63.34	48.12	76.69

Filter Bank Bandwidth and Number of Filters

Next, various filter bank bandwidths and different number of filters are experimented with to view its effect on the classification accuracy of MFCCs and GTCCs, similar to [10]. For MFCCs, the number of filters is typically in the range of

20 to 40. For example, 20 filters are used in [42], 23 filters in [9], 18-24 filters are suggested in [13], 20-24 filters in [45], and 40 filters in [166]. Also, various lower cut-off frequency values have been used in literature such as 20 Hz in [10] and 133.33 Hz in [166]. However, the upper cut-off frequency is often chosen as the Nyquist frequency.

For GTCCs, the filter bank bandwidth and number of filters used in various literature has been summarized in [10]. In their summary, the filter bank bandwidth starts from as low as 20 Hz up to an upper limit 11 kHz and the number of filters range from 20 to 128.

With a sampling frequency of 44100 Hz in this work, the maximum possible upper frequency limit is the Nyquist frequency of 22050 Hz. For both mel and gammatone filter banks, the limits are set as multiples of the sampling frequency with the lower limits as $\left[0, {}^{F_s}/_{N}, {}^{2F_s}/_{N}, {}^{3F_s}/_{N}, {}^{4F_s}/_{N}\right]$ and the upper limits as $\left[{}^{F_s}/_{8}, {}^{F_s}/_{4}, {}^{3F_s}/_{8}, {}^{F_s}/_{2}\right]$.

The average classification accuracy value for different filter bank bandwidths are given in Table 5.6 and Table 5.7 for MFCCs and GTCCs, respectively. For MFCCs, the bandwidth of 258.40 Hz – 16537.5 Hz gives the highest average classification accuracy and 172.27 Hz – 16537.5 Hz for GTCCs. The upper bandwidth limit of 16537.5 Hz is same for both MFCCs and GTCCs and this is because for most of the classes, the dominant frequency components lie below this frequency.

Table 5.6: Average classification accuracy value with various filter bank bandwidths for MFCCs

		$f_h\ (Hz)$			
		5512.5	11025	16537.5	22050
$f_l\ (Hz)$	0	66.09	70.57	71.15	73.05
	86.13	66.53	71.64	73.23	71.25
	172.27	67.31	72.20	72.62	71.55
	258.40	64.93	71.29	**73.89**	73.07
	344.53	66.79	73.79	73.86	73.28

Table 5.7: Average classification accuracy value with various filter bank bandwidths for GTCCs

		$f_h\ (Hz)$			
		5512.5	11025	16537.5	22050
$f_l\ (Hz)$	0	71.55	75.36	77.39	76.69
	86.13	73.39	75.54	77.87	77.60
	172.27	74.63	77.41	**78.83**	76.90
	258.40	75.71	77.43	77.38	76.01
	344.53	76.24	78.53	77.03	75.77

In Figure 5.1(a) and (b), comparison is done on the effect of increasing number of filters on the average classification accuracy of MFCCs and GTCCs, respectively. For MFCCs, experimentation was done with 20 to 40 filters in increments of two filters while for GTCCs experimentation was done with 20 to 96 filters in increments of four filters. For MFCCs, the highest average classification accuracy is at 26 filters and 24 filters for GTCCs. In general, for both features, the overall classification performance was seen to decrease as the number of filters increased.

Fine-Tuned Results with Various Classifiers

The fined-tuned parameter settings for the two features are as follows: MFCCs: $M_1 = 26$, $f_l = 258.4$ Hz, and $f_h = 16537.5$ Hz; GTCCs: $M_2 = 24$, $f_l = 172.27$ Hz, and $f_h = 16537.5$ Hz. Also, of the three ERB filter models considered for GTCCs, Lyon's filter model was shown to give the best results so results using this model only are presented from now on.

For both the features, the feature vector for each frame is 36-dimensional: 12 cepstral coefficients plus the first and second derivatives. The overall feature vector dimension for a signal is $36 \times N_t$, where N_t is the total number of frames in the sound signal, which is different in each case depending on the length of the signal. After data normalization, the feature vector is represented by concatenating the mean and standard deviation for each dimension. As such, the final feature vector is 72-dimensional.

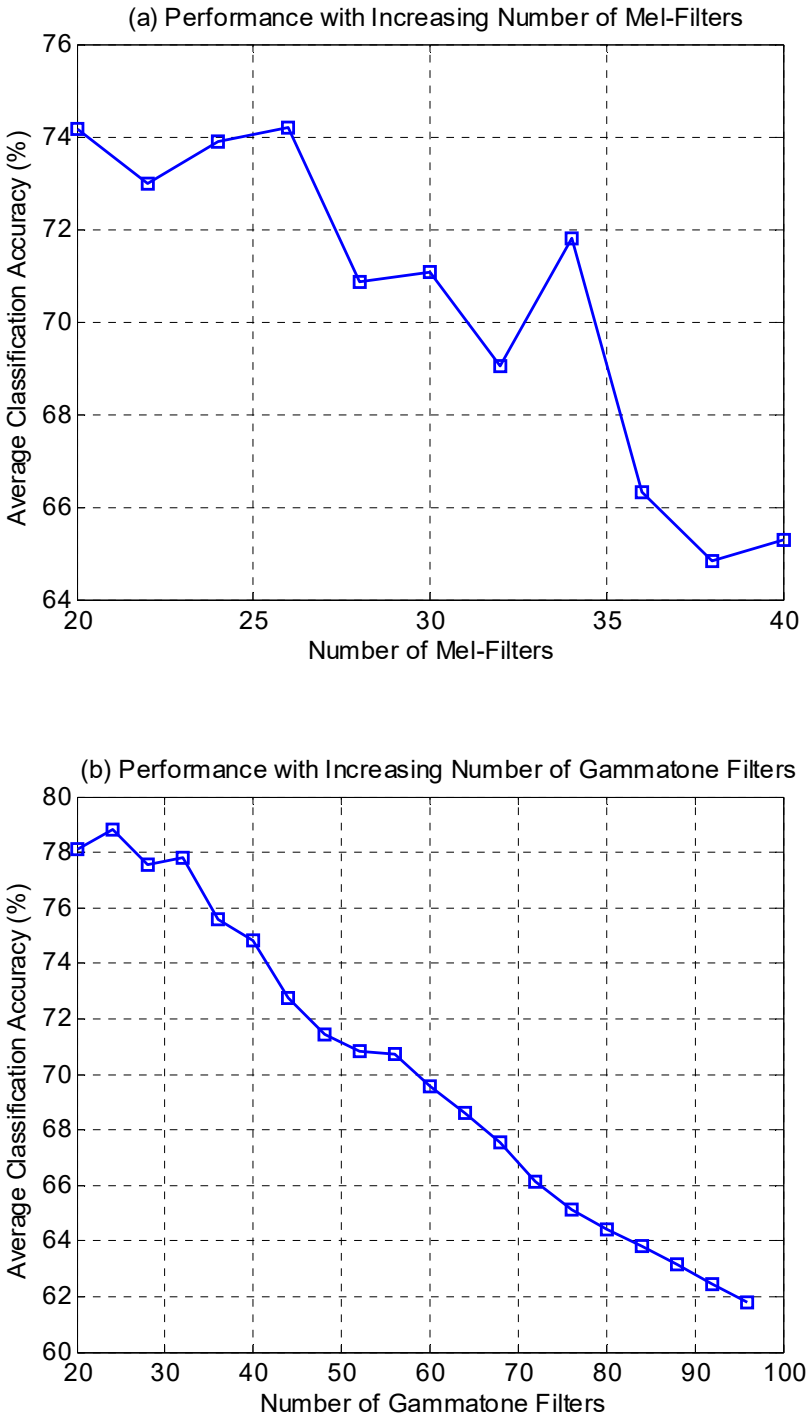


Figure 5.1: Average classification accuracy with increasing number of (a) mel-filters and (b) gammatone filters

The classification accuracy values for MFCCs and GTCCs using log compression and using the fine-tuned frequency range and number of filters is given in Table 5.8 using SVM, *k*NN, and DNN classifiers. When compared to the results given in Table 5.5, the average classification accuracy value for both the cepstral features

show some improvement, increasing from 73.05% to 74.19% for MFCCs and 76.69% to 78.83% for GTCCs with the baseline classifier. The most significant improvement for both features is at 5dB and 0dB SNRs.

Table 5.8: Classification accuracy values for log MFCCs and GTCCs with different classification methods at fine-tuned parameter settings

Classification Method	Log MFCC						Log GTCC					
	Clean	20dB	10dB	5dB	0dB	Ave	Clean	20dB	10dB	5dB	0dB	Ave
OAA-SVM	97.11	92.21	73.32	60.54	47.77	74.19	96.33	94.58	77.78	70.43	55.03	78.83
OAO-SVM	98.16	91.08	75.59	57.74	40.59	72.63	97.64	94.84	80.05	67.80	49.43	77.95
DDAG-SVM	98.16	91.08	75.50	58.01	40.42	72.63	97.64	94.84	77.17	65.62	48.91	76.83
ADAG-SVM	98.16	91.08	75.07	59.58	43.57	73.49	97.64	94.66	77.08	64.92	50.48	76.96
*k*NN	93.18	86.26	67.19	49.96	36.48	66.61	92.13	90.90	69.55	57.57	46.11	71.25
DNN	96.85	90.03	81.36	66.05	50.48	**76.96**	96.85	95.19	81.98	68.24	57.04	**79.86**

In both sets of results, the OAA-SVM classification method gives the best average classification accuracy of the four multiclass SVM classification methods. While there isn't a significant difference in the classification accuracy using the four methods in clean conditions and at high SNRs, the OAA-SVM classification method generally gives better performance at low SNRs. Also, all multiclass SVM classification methods give significantly better classification accuracy than the *k*NN classifier under all noise conditions. However, the DNN classifier gives the best overall classification performance for both the features and is generally more noise robust.

For both the cepstral features, the classification accuracy in clean conditions and at 20dB SNR are greater than 90% using SVM and DNN classification methods. However, the classification accuracy reduces greatly with the addition of noise at 10dB, 5dB, and 0dB SNRs with classification accuracy values of 81.98%, 68.24%, and 57.04%, respectively, with the best overall performing feature, GTCCs, and the best overall performing classifier, DNN.

5.4.2 Linear Cepstral Coefficients

Determining the Optimal Root Value

Next, the effect of root compression on the classification accuracy is examined. The average classification accuracy value at different root values are plotted in Figure 5.2(a) and (b) for MFCCs and GTCCs, respectively, using the baseline classifier. For MFCCs, the average classification accuracy is generally increasing as the root value increases up to the maximum root value of 1 which has been suggested in [13]. As such, experimentation was done beyond this value up to a root value of 3. For both features, there is significant improvement in overall classification performance when compared to the reference classification accuracy given using log compression.

Results using Linear Cepstral Coefficients

Since the best results for both MFCCs and GTCCs are achieved around $\gamma = 1$, the classification accuracy at this setting is considered which is referred as linear cepstrum implying no compression is applied to the filter bank energies. The classification accuracy values for MFCCs and GTCCs using linear compression are given in Table 5.9. Similar to the results using log compression given in Table 5.8, GTCCs once again give the highest average classification accuracy. Generally, there is a marginal decline in classification performance under clean conditions which can be expected since log compression gives better emphasis on the low energy components. However, there is a significant increase in the classification accuracy at 10dB, 5dB, and 0dB SNRs with linear compression.

While the OAA multiclass SVM classification method once again outperforms the other three multiclass SVM classification methods with linear MFCCs, the OAO method is slightly better the OAA method in the case of linear GTCCs. For both the features though, the OAA method gives the best performance of the multiclass SVM classification methods at 5dB and 0dB SNRs. However, once again, the DNN classification method gives the best overall classification performance, and, generally, the most noise robust as well. The classification accuracy peaks at around $\gamma = 1$ which means all cepstrum magnitudes are given equal importance unlike with log compression. The results for the *k*NN classifier are more improved than SVM and DNN classifiers for this reason.

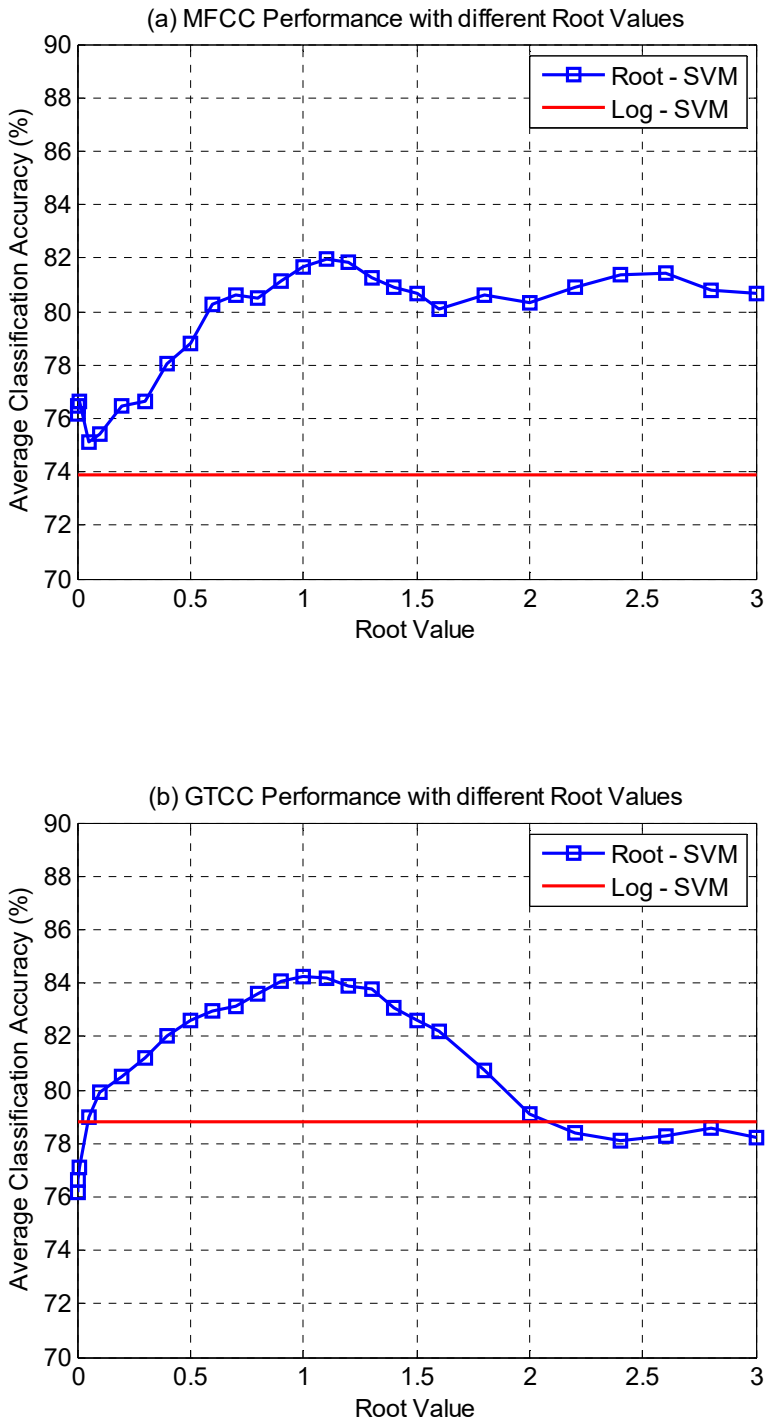


Figure 5.2: Average classification accuracy value for (a) MFCCs and (b) GTCCs with different root values

Table 5.9: Classification accuracy values for linear MFCCs and GTCCs

Classification Method	Linear MFCC						Linear GTCC					
	Clean	20dB	10dB	5dB	0dB	Ave	Clean	20dB	10dB	5dB	0dB	Ave
OAA-SVM	96.06	93.70	84.25	74.98	60.72	81.94	96.85	93.96	87.75	80.93	61.77	84.25
OAO-SVM	96.33	91.60	81.98	69.38	52.76	78.41	97.38	97.29	90.99	80.05	59.41	85.02
DDAG-SVM	96.85	92.04	82.06	68.85	51.88	78.34	97.38	95.98	89.24	78.83	57.66	83.81
ADAG-SVM	96.33	91.86	81.80	67.98	50.66	77.73	97.38	95.98	89.24	78.92	58.01	83.90
*k*NN	94.23	91.78	85.04	76.29	65.09	82.48	95.54	93.96	86.35	80.14	65.53	84.30
DNN	95.28	95.10	88.19	78.65	65.18	**84.48**	95.80	95.63	88.80	81.80	66.49	**85.70**

Log vs Linear

To understand the greater noise robustness of root compression over log compression, in Figure 5.3(a) and (b), the mel cepstrum for a frame using log compression and root compression, $\gamma = 1$, are plotted, respectively. The values are plotted under clean conditions and with the addition of noise at 0dB SNR. The deviation of the noise manipulated root cepstrums from the noise free root cepstrums is much smaller than the deviation of the log cepstrums which explains its greater immunity to noise.

5.4.3 Spectrogram Image Feature

Optimal Number of Blocks

The last baseline feature is the SIF. For the SIF, the spectrogram image is divided into 9×9 blocks and second and third central moments are computed in each block. These values are then concatenated to form the final feature vector which is 162-dimensional. Experimentation was also carried out with $3 \times 3, 5 \times 5,$ and 7×7 blocks but best results were obtained with 9×9 blocks, which was also the maximum that could be experimented with due to limitations in the length of the sound signal and the length of the spectrogram image as a result. The classification accuracy values with different number of blocks for linear and log spectrograms are given in Table 5.10 using the baseline classifier. In general, the average classification performance increases as the number of blocks increases. The chosen block size is also consistent with that used in [2].

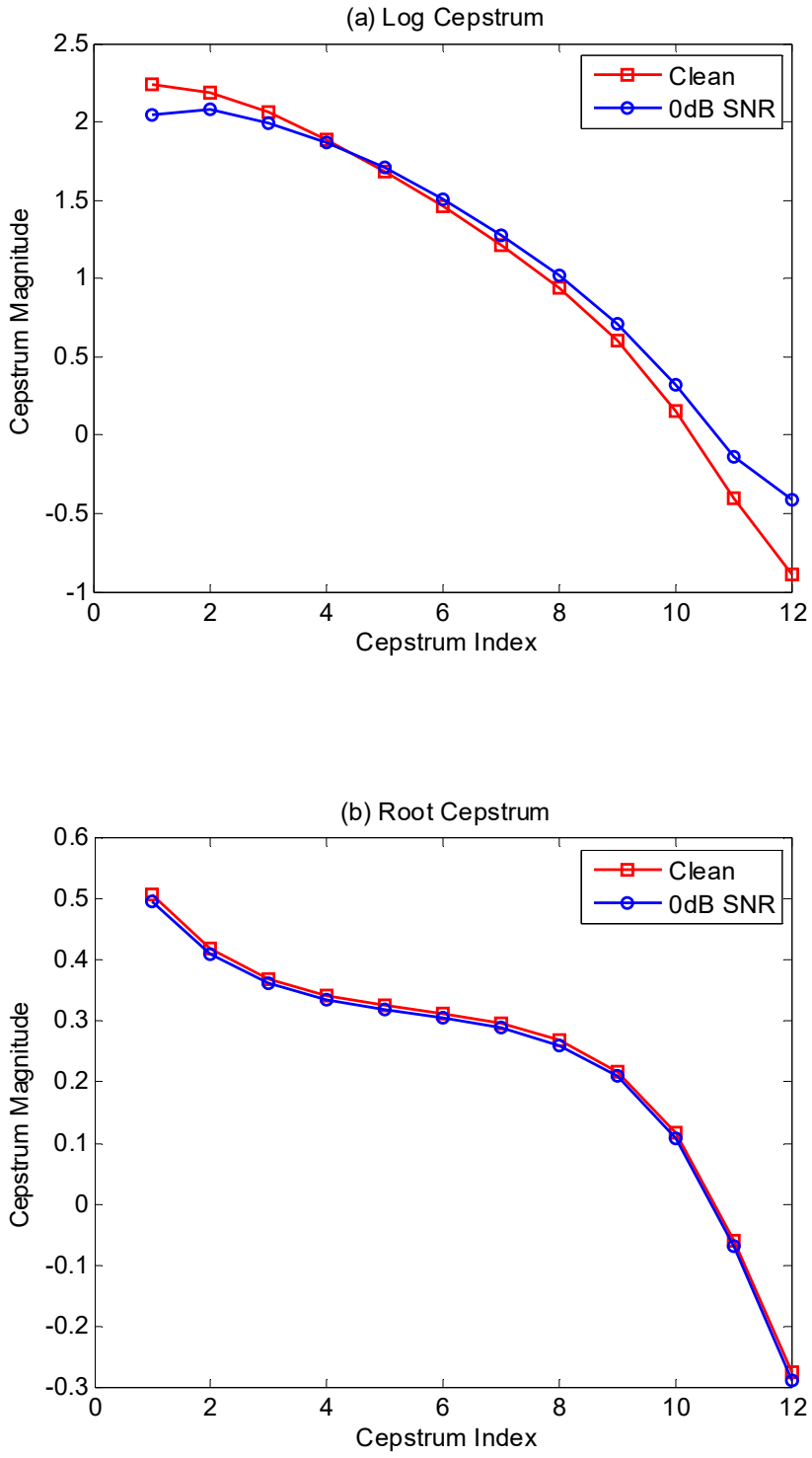


Figure 5.3: Comparison of the effect of (a) log compression and (b) root compression on mel cepstrum with the addition of noise

Table 5.10: Classification accuracy values for SIF with different sized blocks

No. of blocks	Linear SIF						Log SIF					
	Clean	20dB	10dB	5dB	0dB	Ave	Clean	20dB	10dB	5dB	0dB	Ave
3 × 3	82.68	82.59	76.55	66.14	40.68	69.73	93.70	63.95	51.44	45.32	36.83	58.25
5 × 5	90.81	90.20	85.83	65.44	38.06	74.07	95.01	72.53	45.93	38.32	31.41	56.64
7 × 7	89.76	89.76	84.78	67.98	38.58	74.17	94.75	69.73	48.73	39.72	31.93	56.97
9 × 9	91.60	91.34	88.80	67.19	40.51	**75.89**	93.70	70.60	54.59	44.36	36.22	**59.90**

Results with Various Classifiers

For both the spectrogram representations, the classification accuracy values using the four multiclass SVM classification methods and the *k*NN and DNN classifiers are given in Table 5.11. For the multiclass SVM classification methods, the OAA method once again gives the best overall results with both linear and log SIF. While the overall classification accuracy using OAA is significantly better than the other methods for linear SIF, the classification performance with log SIF is much more even. The OAA method is once again the most noise robust, giving the highest performance at 10dB, 5dB, and 0dB SNRs. However, the *k*NN and the DNN classifiers give a better overall classification performance than the multiclass SVM classification methods for linear SIF and the DNN classifier once again gives the best classification performance. Interestingly, the classification performance reverses with log SIF with both *k*NN and DNN classifiers giving lower overall classification performance than the SVM methods. As with linear ceptrums, the *k*NN classification method is seen to be more effective with the linear spectrograms.

When compared to the cepstral features, at 84.27%, the linear SIF gives significantly better overall classification performance than conventional or log compressed MFCCs and GTCCs which gave a best overall classification accuracy of 76.96% and 79.86%, respectively. The classification values using linear SIF are lower under clean conditions but significantly better at 10dB, 5dB, and 0dB SNRs, making the linear SIF more noise robust. However, the linear SIF gives marginally lower average classification performance than linear MFCCs and also linear GTCCs, which, with an average classification accuracy of 85.70%, is the best performing baseline feature.

Table 5.11: Classification accuracy values for SIF with 9×9 blocks using different classification methods

Classification Method	Linear SIF						Log SIF					
	Clean	20dB	10dB	5dB	0dB	Ave	Clean	20dB	10dB	5dB	0dB	Ave
OAA-SVM	91.60	91.34	88.80	67.19	40.51	75.89	93.70	70.60	54.59	44.36	36.22	**59.90**
OAO-SVM	89.76	78.92	69.12	55.38	37.45	66.12	95.54	72.88	51.27	42.17	31.58	58.69
DDAG-SVM	87.93	73.58	60.89	51.36	32.63	61.28	95.54	71.30	49.61	41.56	31.58	57.92
ADAG-SVM	87.93	80.23	69.90	55.21	34.38	65.53	95.28	72.27	49.96	41.29	32.90	58.34
*k*NN	86.88	86.00	83.55	79.27	58.27	78.79	90.81	63.08	35.17	27.12	21.26	47.49
DNN	93.18	92.91	91.51	83.03	60.72	**84.27**	87.93	65.00	42.78	31.41	24.76	50.38

5.5 Results using Proposed Spectrogram Image Features

In this section, the performance of the proposed spectrogram image features are presented. The two proposed spectrogram image features are the RSIF and the SITF.

5.5.1 Reduced Spectrogram Image Feature

Optimal Number of Blocks

For the RSIF, similar to the SIF, the spectrogram image is divided into 9×9 blocks and second and third central moments are computed in each block. However, the feature dimension is reduced using the mean and standard deviation of the central moment values along the rows and columns of the blocks, as illustrated in Figure 3.9. This results in a 72-dimensional final feature vector. Once again, experimentation was done with $3 \times 3, 5 \times 5,$ and 7×7 blocks as well and the classification accuracy values for these block sizes for linear and log spectrograms are given in Table 5.12 using the baseline classifier. Interestingly, the best overall classification accuracy was achieved using 3×3 blocks for the log representation, however, the linear spectrogram representation once again gives significantly better performance and best using 9×9 blocks.

Table 5.12: Classification accuracy values for RSIF with different sized blocks

No. of blocks	Linear RSIF						Log RSIF					
	Clean	20dB	10dB	5dB	0dB	Ave	Clean	20dB	10dB	5dB	0dB	Ave
3 × 3	83.73	82.85	76.12	65.09	44.79	70.52	95.54	65.97	51.88	43.83	35.78	**58.60**
5 × 5	92.91	92.04	88.28	77.52	46.46	79.44	96.06	70.17	45.93	37.88	32.90	56.59
7 × 7	92.13	91.78	88.01	77.17	49.96	79.81	97.38	70.17	48.12	37.97	31.15	56.96
9 × 9	92.13	92.04	89.33	78.57	53.37	**81.08**	96.06	72.00	50.48	38.93	31.32	57.76

Results with Various Classifiers

The classification accuracy values for linear RSIF and log RSIF with the different classification methods are given in Table 5.13. While the average classification accuracy using the proposed RSIF is slightly lower for the log representation when compared to the SIF, the average classification accuracy with the linear representation, which gives the best results, is significantly higher, increasing from 84.27% with linear SIF to 87.56% with linear RSIF. The RSIF method has the added advantage of a feature vector which is 2.25 times smaller in dimension. As such, the proposed method can be said to be much more effective for its dimension.

Table 5.13: Classification accuracy values for RSIF with 9 × 9 blocks using different classification methods

Classification Method	Linear RSIF						Log RSIF					
	Clean	20dB	10dB	5dB	0dB	Ave	Clean	20dB	10dB	5dB	0dB	Ave
OAA-SVM	92.13	92.04	89.33	78.57	53.37	81.08	96.06	72.00	50.48	38.93	31.32	57.76
OAO-SVM	92.13	86.70	82.33	72.79	48.29	76.45	97.11	73.05	50.22	38.93	30.80	58.02
DDAG-SVM	91.86	87.40	82.50	66.67	44.97	74.68	97.38	72.62	48.91	38.15	30.62	57.53
ADAG-SVM	90.81	86.70	82.15	71.92	48.29	75.98	97.90	74.45	50.57	39.55	31.50	58.79
*k*NN	87.93	87.23	83.90	78.48	56.61	78.83	93.44	64.57	39.28	30.27	24.23	50.36
DNN	94.23	93.88	93.79	89.76	66.14	**87.56**	95.01	76.64	51.01	40.07	33.77	**59.30**

When compared to the conventional cepstral features, that is, log-compressed cepstrums, the average classification accuracy using RSIF is significantly better than MFCCs and GTCCs. While the classification accuracy under clean conditions is lower, the RSIF generally gives better classification performance in the presence of noise. In addition, unlike the SIF, the RSIF gives a higher overall classification performance than both linear MFCCs and linear GTCCs.

Log vs Linear

A time-frequency image represents two-dimensional data which makes it more useful for feature extraction when compared to the one-dimensional data available in time-domain and frequency-domain representation of the signal on its own. The log spectrogram approach gives the highest classification accuracy in clean conditions which can be expected since taking log power reveals the details in the low power frequencies unlike the linear spectrogram approach where only the dominant power frequencies are shown. This can be visualized in the linear spectrogram and log spectrogram images in Figure 3.8(a) and (c), respectively. However, the performance of the two representations reverses with the addition of noise. The noise is more diffuse than the sound signal and its power affects most of the frequencies in the log grayscale image as shown in Figure 3.8(d). For the linear representation, the strong peaks of the sound are larger than the noise and remain largely unaffected with the addition of noise as can be seen in Figure 3.8(b).

5.5.2 Spectrogram Image Texture Feature

Optimal Parameter Settings

For obtaining the SITF, the GLCM method of texture analysis is firstly applied to the spectrogram images. Since the linear spectrogram representation has shown to be more effective for the SIF and RSIF, only this representation is considered from now on. The feature vector is then formed by concatenating the columns of the matrix. In preliminary experiments, the classification accuracy with increasing values of N_g was compared. The average classification accuracy was seen to decrease as N_g increased, therefore, for all the experiments that follow, $N_g = 2$ is used, which gave the highest average classification accuracy.

Two other experiments are performed the aim of which is to:

- compare the classification accuracy using feature vectors formed from application of GLCM analysis at angles of $0°, 45°, 90°, \text{and } 135°$ and then with combined feature vector, and
- compare the classification accuracy with increasing number of frequency bands.

The results using the baseline classifier for the first set of experiments are given in Table 5.14 with $N_g = 2$ and $d = 1$. Comparing the average classification accuracy, for the individual feature vectors, the best average classification accuracy is achieved with analysis an angle of $45°$ while the combined feature vector gives marginally better classification accuracy. In this experiment, the spectrogram image is not divided into subbands before feature extraction. Therefore, the feature vector dimension when analyzing at individual angles is $N_g^2 = 4$ and $4N_g^2 = 16$ when the feature vector from the four angles are combined. As such, while the feature vector dimension has quadrupled when combined, there isn't a considerable increase in the classification accuracy in comparison to the best performing individual feature vector. However, the classification performance at this stage is far below the best performing features, linear GTCCs and RSIF, as seen so far.

Table 5.14: Classification accuracy values using the SITF – individual and combined feature vectors

Angle	Clean	20dB	10dB	5dB	0dB	Ave
0°	84.78	84.60	77.69	68.24	49.34	72.93
45°	82.15	81.98	80.23	75.33	56.61	**75.26**
90°	76.12	76.12	74.45	70.34	50.39	69.48
135°	81.36	81.28	78.57	72.00	54.42	73.53
All Angles	86.09	85.74	81.45	74.89	55.03	**76.64**

The next experiment looks at the effect on the classification accuracy of performing GLCM analysis with increasing number of frequency bands. The spectrogram image is now divided into blocks of horizontal sections with equal number of frequency

bins in each subband. The GLCM is computed in each subband which are then concatenated into one matrix. This matrix is then concatenated into a column vector which forms the final feature vector. The number of pixels in the spectrogram image along the vertical, or frequency axis, is $N/2 = 256$, therefore, various number of frequency bands, N_b, from 1 to 256 can be experimented with. Experimentation was performed with $N_b = 1, 2, 4, 8, 16, 32, 64, \text{and } 128$ at a time. The results presented in Table 5.15 use the baseline classifier with feature vector combined from all four angles.

Table 5.15: Classification accuracy values using SITF (combined feature vector) – effect of increasing number of subbands

N_b	Clean	20dB	10dB	5dB	0dB	Ave
1	86.09	85.74	81.45	74.89	55.03	76.64
2	85.04	84.51	81.98	75.42	57.83	76.96
4	85.04	84.34	82.59	77.17	59.58	77.74
8	83.99	83.99	82.33	77.34	59.49	77.43
16	86.09	86.09	84.16	82.15	62.38	80.17
32	87.93	87.66	86.79	86.00	69.03	83.48
64	**90.29**	**89.68**	**89.59**	**87.75**	**73.40**	**86.14**
128	88.71	88.63	88.10	85.83	72.27	84.71

While there isn't a significant change in the classification accuracy with increasing values of N_b at lower values of N_b, there is notable increase in the classification accuracy from $N_b = 16$ onwards with the most improved results at 5dB and 0dB SNRs. The highest classification accuracy under all noise conditions is at $N_b = 64$, therefore, giving the best overall classification performance as well.

However, the disadvantage of the proposed method is its high computational cost. The SITF dimension using subband analysis and with the combined feature vector from all four angles can be given as $4\left(N_b \times N_g^2\right)$. With $N_b = 64$, where the highest classification accuracy is achieved, the feature vector dimension is 1024, which is about 6.32 times more than the SIF and 14.22 times more than MFCCs, GTCCs, and the RSIF. The subband analysis technique was also applied to feature vector

from each of the four angles. In general, it was observed that as N_b increased in value, the difference in the classification accuracy between individual feature vectors and the combined feature vector got minimal. Table 5.16 gives the classification accuracy values using feature vectors from each of the four angles considered with $N_b = 64$. Best results were once again achieved with features extracted from analysis at an angle of $45°$. While the individual feature vectors give slightly lower classification accuracy than the combined feature vector, these can be considered more effective since the feature vector dimension is much lower, reduced by 4 to 256. As such, from here on, all results are given using GLCM analysis at an angle of $45°$.

Table 5.16: Classification accuracy values using SITF with individual feature vectors at the optimal number of subbands

Angle	Clean	20dB	10dB	5dB	0dB	Ave
0°	88.45	88.45	87.84	84.95	69.29	83.80
45°	89.76	89.41	89.33	87.66	71.92	**85.62**
90°	88.45	88.01	87.93	86.44	70.78	84.32
135°	88.71	88.71	88.36	86.44	71.13	84.67

Experimentation was also done with increasing values of d and it was observed that while increasing d from 1 to 2 increases the average classification accuracy for lower values of N_b, the difference between the two sets of results got smaller as the value of N_b increased. Eventually, the average classification accuracy with $d = 1$ surpassed those at $d = 2$, and, at $N_b = 64$, the highest classification accuracy was still achieved using $d = 1$.

Results with Various Classifiers

The classification accuracy values for the SITF at the optimal parameter settings using the different classification methods are given in Table 5.17. Looking at the multiclass SVM classification methods, the OAA method once again gives the most noise robust and best overall classification performance. The classification performance of the *k*NN classifier is comparable to OAO, DDAG, and ADAG

multiclass SVM classification methods but, once again, the DNN classifier is the best of the lot, significantly outperforming the other classifiers under each noise condition.

Table 5.17: Classification accuracy values for SITF with different classifiers

Classification Method	Clean	20dB	10dB	5dB	0dB	Ave
OAA-SVM	89.76	89.41	89.33	87.66	71.92	85.62
OAO-SVM	88.98	88.98	88.19	83.46	60.37	81.99
DDAG-SVM	85.30	85.39	83.64	77.25	54.86	77.29
ADAG-SVM	87.14	87.40	85.30	83.11	61.07	80.80
*k*NN	81.36	81.71	81.36	80.84	74.80	80.02
DNN	95.28	95.28	94.93	94.31	80.84	**92.13**

When compared to results using log MFCCs and GTCCs, the proposed SITF is not able to match the classification accuracy under clean conditions but gives significantly better performance in the presence of noise, especially as the SNR decreases. With an overall classification accuracy of 92.13%, the SITF also outperforms linear GTCCs, the best performing baseline feature, at 85.70%. The classification accuracy values are comparable under clean conditions and at 20dB SNR but an improvement of 6.74%, 12.51%, and 14.35% is achieved at 10dB, 5dB, and 0dB SNRs, respectively.

The overall classification performance is also higher than the RSIF by 4.57%. The classification performance is comparable under clean conditions and at 20dB and 10dB SNRs but significantly better classification accuracy is achieved at 5dB and 0dB SNRs, increasing from 89.76% to 94.31% and 66.14% to 80.84%, an increase of 4.55% and 14.70%, respectively. Therefore, the key advantage of the SITF over the features presented previously is its greater robustness at low SNRs. To ensure that this improvement wasn't simply because of the different method of spectrogram image division before feature extraction, the frequency subband analysis method was applied to the SIF and RSIF but there wasn't any significant change in the results.

5.6 Results using Proposed Cochleagram Image Features

In this section, the classification performance of all the spectrogram image features, SIF, RSIF, and SITF, but using cochleagram image for feature extraction, are presented. The SIF, RSIF, and SITF are, therefore, referred as CIF, RCIF, and CITF, respectively.

5.6.1 Results for CIF, RCIF, and CITF with All ERB Models

Cochleagram feature extraction follows the same procedure as the spectrogram images but now using a cochleagram image. To get the same image resolution as the spectrogram images, the number of gammatone filters, M_2, is set to 256 with the same window size, $N = 512$. The classification accuracy values for CIF, RCIF, and CITF are given in Table 5.18, Table 5.19, and Table 5.20, respectively. The results in each case are presented using the three ERB filter models using the baseline classifier.

Table 5.18: Classification accuracy values for CIF using the three ERB filter models

ERB Filter Model	CIF					
	Clean	20dB	10dB	5dB	0dB	Ave
Glasberg and Moore [153]	92.13	91.78	90.73	85.74	63.08	84.69
Lyon [154]	91.60	91.25	90.46	83.38	58.88	83.11
Greenwood [155]	93.18	93.09	92.21	89.06	63.95	**86.30**

Table 5.19: Classification accuracy values for RCIF using the three ERB filter models

ERB Filter Model	RCIF					
	Clean	20dB	10dB	5dB	0dB	Ave
Glasberg and Moore [153]	94.75	94.58	94.14	89.68	65.44	87.72
Lyon [154]	95.01	94.40	93.35	89.59	65.44	87.56
Greenwood [155]	94.75	94.75	94.58	91.69	69.38	**89.03**

The average classification accuracy values for CIF and RCIF with all the ERB filter models show significant improvement when compared to SIF and RSIF, respectively. The highest average classification accuracy for both CIF and RCIF is achieved using Greenwood [155] parameters. As such, the average classification accuracy value increases from 75.89% with SIF to 86.30% with CIF, an increase of 10.41%, and from 81.08% with RSIF to 89.03% with RCIF, an increase of 7.95%.

Furthermore, for the CITF, the highest average classification accuracy is achieved using Glasberg and Moore [153] parameters, as per the results in Table 5.20. There is once again an improvement in the average classification accuracy when compared to the spectrogram based features, increasing from 85.62% with SITF to 89.24% with CITF, an increase of 3.62%.

Table 5.20: Classification accuracy values for CITF using the three ERB filter models

ERB Filter Model	CITF					
	Clean	20dB	10dB	5dB	0dB	Ave
Glasberg and Moore [153]	92.65	92.65	92.21	90.38	78.30	**89.24**
Lyon [154]	92.13	91.78	91.34	89.41	80.75	89.08
Greenwood [155]	91.86	91.78	91.78	89.85	78.04	88.66

5.6.2 Results for Best ERB Model with DNN

The classification performance of the OAA multiclass SVM classification method has generally been better than the other three multiclass classification methods and the *k*NN classifier so far. As such, in this instance, the classification performance is only compared against the DNN classifier, the results for which are given in Table 5.21 for the three cochleagram image features with the best performing ERB model in each case. The DNN classifier once again outperforms the baseline classifier under all noise conditions for all three features. Unlike the SIF, the overall classification performance of CIF is better than linear GTCCs, the best performing baseline feature, and, as with spectrogram image feature extraction, the reduced feature method, RCIF, gives better classification performance over the CIF. Also,

similar to the SITF, the CITF gives a better overall classification performance than CIF and RCIF and also the most noise robust. While there isn't a huge difference between the classification accuracy values under clean and high SNR conditions, the CITF is seen to be more effective at low SNRs, 0dB SNR, in particular.

Table 5.21: Classification accuracy values for CIF, RCIF, and CITF with the best performing ERB filter model using OAA-SVM and DNN classifiers

	OAA-SVM						DNN					
	Clean	20dB	10dB	5dB	0dB	Ave	Clean	20dB	10dB	5dB	0dB	Ave
CIF	93.18	93.09	92.21	89.06	63.95	86.30	94.75	94.66	93.79	90.55	70.87	88.92
RCIF	94.75	94.75	94.58	91.69	69.38	89.03	96.06	95.54	95.19	92.39	72.70	90.38
CITF	92.65	92.65	92.21	90.38	78.30	89.24	95.80	95.63	95.45	95.19	88.54	**94.12**

In addition, the improvement in the classification accuracy increases as the SNR decreases. From SIF to CIF, the classification accuracy value increases 1.57%, 1.75%, 2.28%, 7.75%, and 10.15% under clean conditions and at 20dB, 10dB, 5dB, and 0dB SNRs, respectively, with an overall improvement of 4.65%. This shows that while the classification accuracy value under all noise conditions has improved, the most improved results are at low SNRs, 5dB and 0dB SNRs, in particular. Similarly, from RSIF to RCIF, the classification accuracy value increases 1.83%, 1.66%%, 1.40%, 2.63%, and 6.56% under clean conditions and at 20dB, 10dB, 5dB, and 0dB SNRs, respectively, with an overall improvement of 2.82%. Finally, from SITF to CITF, the improvement in classification accuracy is 0.52%, 0.35%, 0.52%, 0.88%, and 7.7% under clean conditions and at 20dB, 10dB, 5dB, and 0dB SNRs, respectively, with an overall improvement of 1.99%.

Therefore, all the time-frequency image features show improvement in classification accuracy under all noise conditions when using a cochleagram image for feature extraction instead of the spectrogram image. While the improvement in the overall classification performance is more with the baseline classifier than with DNN, the classification performance using DNN is still better. Unlike CIF, the improvement in classification accuracy value is generally much more even for RCIF and CITF and

the improvement in the average classification accuracy lower. However, CITF can still be considered the most noise robust feature with a classification accuracy of 95.19% and 88.54% at 5dB and 0dB SNRs, respectively.

5.7 Results using Proposed Feature Combinations

In this section, results using a combination of features are presented. Two sets of feature combinations are considered in this work: cepstral + time and frequency domain features and cepstral + time-frequency image features. Also, so far the performance analysis of the classifiers has been limited to individual features. This section also looks at the classification performance of the various classifiers with feature vector combination.

5.7.1 Cepstral + Time and Frequency Domain Features

First, a combination of cepstral and time and frequency domain features is considered, which has been the norm in a number of other similar research. Since linear GTCCs are the best performing cepstral feature, only linear GTCCs are considered here. However, various time and frequency domain features are considered which are as follows: ZCR, STE, SBE, SC, BW, and SR. The average classification accuracy value for linear GTCCs in combination with these time and frequency domain features is plotted in Figure 5.4 using the baseline classifier. The average classification accuracy using linear GTCCs only is used as reference here. The combination with ZCR, STE, and SBE, combined one at a time, gives some improvement in the average classification performance with the highest result using linear GTCC + SBE at 87.16%. However, there is hardly any change in the classification performance with the inclusion of SC while the inclusion of BW and SR reduce the average classification accuracy. As such, it can be deduced that not all features have a positive impact on the classification performance with feature vector combination.

In addition, while the inclusion of all time and frequency domain features at once gives improvement in the classification performance, it is lower than the average classification accuracy of linear GTCC + SBE. This could largely be attributed to the inclusion of ineffective features in SC, BW, and SR. As such, the final plot in the

figure is with the inclusion of ZCR, STE, and SBE, which were the only time and frequency domain features to give improvement in average classification performance when combined individually with linear GTCCs. At 88.14%, this combination is seen to give the best average classification accuracy, significantly better than the average classification accuracy of 84.25% with linear GTCCs on its own.

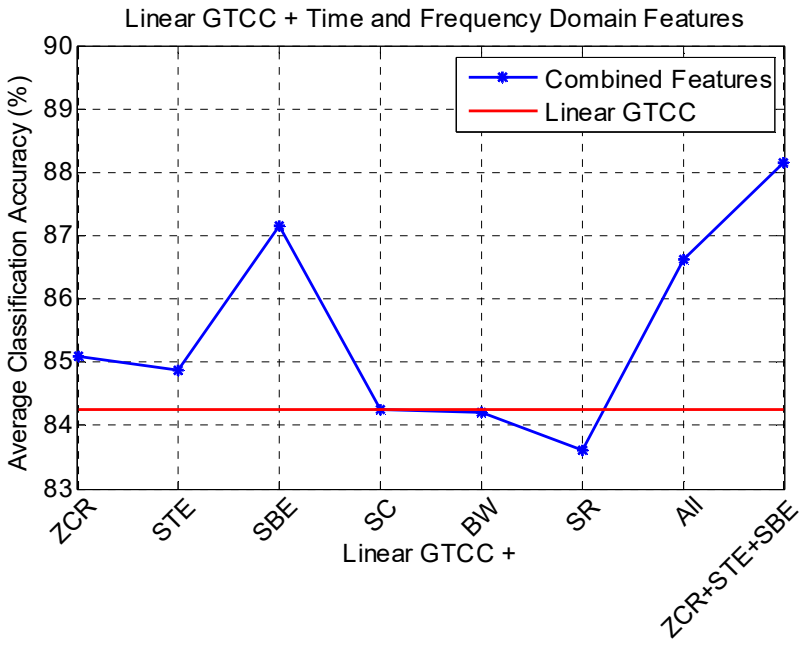


Figure 5.4: Average classification accuracy of linear GTCC + time and frequency domain features

5.7.2 Cepstral + Time-Frequency Image Features

Next, the classification performance using a combination of cepstral and time-frequency image features is presented. Once again, for the cepstral features, linear GTCCs are considered being the best performing cepstral feature. However, log GTCCs are also considered here so that a comparison of log and linear cesptrums could be done in feature vector combination. For the time-frequency image features, only cochleagram derived features are considered since these have been shown to be more robust than spectrogram image derived features.

The classification accuracy values using a combination of GTCCs and cochleagram image features, with the best overall performing ERB model used in each case, is given in Table 5.22 using the baseline classifier. The average classification accuracy values for all cochleagram image features show improvement when combined with log and linear GTCCs. Compared to the average classification accuracy of CIF, RCIF, and CITF, when combined with log GTCCs, the improvement is 3.94%, 2.69%, and 2.57%, respectively. Similarly, the improvement is 5.23%, 3.55%, and 4.06% for CIF, RCIF, and CITF when combined with linear GTCCs, respectively.

Table 5.22: Classification accuracy values for log and linear GTCCs in combination with cochleagram image features

	Log GTCCs +						Linear GTCCs +					
	Clean	20dB	10dB	5dB	0dB	Ave	Clean	20dB	10dB	5dB	0dB	Ave
CIF	96.33	95.98	93.88	91.69	73.32	90.24	96.06	95.98	95.28	93.35	76.99	91.53
RCIF	97.38	96.50	94.40	93.09	77.25	91.72	97.64	97.38	96.59	91.51	79.79	92.58
CITF	96.33	95.28	94.66	92.39	80.40	**91.81**	96.59	96.59	95.36	94.23	83.73	**93.30**

The performance when combined with log GTCCs is relatively good considering the relatively poor performance of log GTCCs against linear GTCCs. However, the improvement in classification performance for all three cochleagram features is more with linear GTCCs than log GTCCs. Therefore, feature combination with linear GTCCs can be considered more superior than with log GTCCs. In addition, CIF combined with linear GTCCs gives the most improved results. However, CITF is once again the best performing feature with an average classification accuracy of 93.30% when combined with linear GTCCs. In addition, this combination also gives the most noise robust performance with a classification accuracy of 94.23% and 83.73% at 5dB and 0dB SNRs, respectively.

Furthermore, the combination of cepstral and time-frequency image features is seen to be more effective than the combination of cepstral and time and frequency domain features. Looking at the best feature combination in each case, the average

classification accuracy with linear GTCC + ZCR + STE + SBE is 88.14% and 93.30% with linear GTCC + CITF, a difference on 5.16%.

5.7.3 Classifier Performance with Feature Combination

While the classification accuracy values of the different classifiers have been compared for a number of individual features, in Table 5.23, the performance is compared with feature vector combination. Only the best performing feature combination, linear GTCC + CITF, is considered here and results are given using the four multiclass SVM classification methods and the *k*NN and DNN classifiers. The OAA multiclass SVM classification method is once again seen to more noise robust with a better overall classification performance than the OAO, DDAG, and ADAG methods. The *k*NN classifier is seen to be the least effective with feature vector combination. However, the DNN classifier once again outperforms all classifiers with the highest classification accuracy under each noise condition and an average classification accuracy of 96.06%, 2.76% more than the baseline classifier. The most improved results are at 0dB SNR, with an improvement of 7.52% over the baseline classifier.

Table 5.23: Classification accuracy values for linear GTCCs + CITF with different classification methods

Classification Method	Linear GTCCs + CITF					
	Clean	20dB	10dB	5dB	0dB	Ave
OAA-SVM	96.59	96.59	95.36	94.23	83.73	93.30
OAO-SVM	94.23	94.49	93.44	91.34	81.10	90.92
DDAG-SVM	94.23	94.23	93.00	90.73	80.93	90.62
ADAG-SVM	95.80	95.71	94.49	91.86	81.54	91.88
*k*NN	83.20	82.41	82.06	81.45	78.04	81.43
DNN	97.90	97.81	97.64	95.71	91.25	**96.06**

5.8 Further Analysis

The proposed method of feature extraction using the GLCM gives the most noise robust performance and also the best overall classification performance with

spectrogram feature extraction, cochleagram feature extraction, and when combined with linear GTCCs. The peak of the filter bank energies play a key role in characterizing a sound signal which is demonstrated by the superior performance of both the cepstral features under clean conditions. However, the conventional log compression can produce high variations in the output for low energy components [167] which explains its poor performance as the SNR decreases. While the introduction of linear cepstrums improved the noise robustness, the proposed methods give a far superior performance at low SNRs.

In addition, for features extracted from the linear spectrograms, which have been shown to be more noise robust than log spectrograms, the results achieved using the proposed features, RSIF and SITF, are better than the SIF method of data representation given in [2]. Significant improvement in the classification performance was also achieved by using a cochleagram image for feature extraction over the spectrogram image. The combination of linear GTCCs and cochleagram image features also gave some improvement in classification performance with best classification accuracy values of 97.90%, 97.81%, 97.64%, 95.71%, and 91.25% under clean conditions and at 20dB, 10dB, 5dB, and 0dB SNRs, respectively. All these values are marginally to significantly higher than in [5], a related work the results for which are summarized in section 2.5.2. In addition, the number of classes in this work is one more than in [5] with 66.67% of data used for training when compared to 70% in [5]. As such the classification task in this work can be considered slightly more challenging. While a more noise robust performance is achieved in this work, it is difficult to conclusively say that the techniques presented here are better due to the variations in sound and noise databases. However, the techniques proposed in this work have been shown to outperform a number of baseline methods.

Furthermore, for various individual and combined features, the DNN classifier has been seen to outperform the SVM classifier in terms of overall classification performance and noise robustness. The classification accuracy results for the individual cepstral and time-frequency image features also have some similarity to the results in [106]. For example, in [106], for MFCCs, the improvement in classification performance from SVM to DNN is -9.0%, 20.7%, 22.9%, and 8.5% under clean conditions and at 20dB, 10dB, and 0dB SNRs, respectively, with an

improvement of 10.8% in the average classification performance. In our work, for linear GTCCs, the best performing cepstral feature, the improvement in classification accuracy over the baseline classifier is -1.05%, 1.67%, 1.05%, 0.87%%, 4.72% under clean conditions and at 20dB, 10dB, 5dB, and 0dB SNRs, respectively, with an improvement of 1.45% in the average classification accuracy. While our work does not achieve as much improvement in classification performance as in [106], it should be noted that the evaluation task in [106] was identical to [109] and the results for MFCC-SVM were taken from [109]. It is understood that linear SVM is used in [109] using the OAO multiclass classification method. In our experimentation in [16], the classification performance using nonlinear SVM and OAA multiclass classification method were determined to be better than linear SVM and OAO multiclass classification method, respectively, which could explain the lesser improvements in classification performance in our work than in [106].

For the SAI features, the improvement in classification performance from SVM to DNN in [106] is 1.87%, 1.80%, 8.07%, 9.40% under clean conditions and at 20dB, 10dB, and 0dB SNRs, respectively, with an improvement of 5.28% in the average classification performance. For the CITF, the best performing time-frequency image feature in our work, the improvement in classification performance is 3.15%, 2.98%, 3.24%, 4.81%, and 10.24% under clean conditions and at 20dB, 10dB, 5dB, and 0dB SNRs, respectively, with an improvement of 4.88% in the average classification performance. As such, the improvement in the classification performance for the CITF compares favorably with the SAI features in [106]. However, results for feature vector combination and the training and evaluation times have not been reported in [106].

5.8.1 Interclass Classification

While overall classification accuracy values have been presented so far, to understand the classification performance between classes, the classification and misclassification values of classes are presented here. The confusion matrix for the CITF, the best performing individual feature, under clean conditions and in the presence of noise at 0dB SNR is given in Table 5.24 and Table 5.25, respectively, using the baseline classifier. The values in the confusion matrix are given in

percentage as *number of correctly (or incorrectly) classified samples* divided by *number of test samples in the class*. The rows in the confusion matrix denote the classes that are intended to be classified while the classified results are given in the columns.

For example, for the confusion matrix under clean conditions given in Table 5.24, 96.67% of the test samples from *alarms* were correctly classified while the remaining 3.33% were misclassified into *children voices*. *Dog barking*, *footsteps*, and *glass breaking* also have misclassification in one class only while *gunshots*, *horn*, *machines*, and *phone rings* are the best performing classes with no misclassifications. *Children voices* and *construction* are the worst performing classes with a classification accuracy of 70% and 83.33%, respectively, with both classes also having multiple misclassifications. In addition, there is only one-sided confusion between *footsteps* and *dog barking* whereby test samples from *footsteps* are misclassified into *dog barking* but not vice-versa. *Alarms*, *construction*, *dog barking*, and *glass breaking* have two-sided confusion with *children voices* whereby test samples from each of these classes is misclassified into *children voices* and vice-versa.

Table 5.24: Confusion matrix for test samples under clean conditions using CITF

	Alarms	Children Voices	Construction	Dog barking	Footsteps	Glass breaking	Gunshots	Horn	Machines	Phone rings
Alarms	96.67	3.33	0	0	0	0	0	0	0	0
Children voices	3.33	70.00	5.00	11.67	6.67	1.67	0	1.67	0	0
Construction	0	6.67	83.33	0	0	6.67	0	0	0	3.33
Dog barking	0	3.57	0	96.43	0	0	0	0	0	0
Footsteps	0	0	0	1.75	98.25	0	0	0	0	0
Glass breaking	0	5.00	0	0	0	95.00	0	0	0	0
Gunshots	0	0	0	0	0	0	100.00	0	0	0
Horn	0	0	0	0	0	0	0	100.00	0	0
Machines	0	0	0	0	0	0	0	0	100.00	0
Phone Rings	0	0	0	0	0	0	0	0	0	100.00
Overall Classification Accuracy = 92.65%										

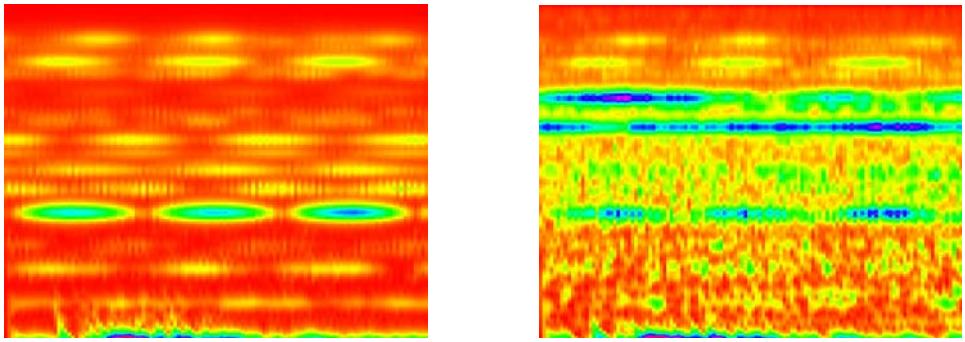
Looking at the confusion matrix at 0dB SNR, Table 5.25, all classes now have misclassifications when compared to only six classes which had misclassification(s) under clean conditions. Once again, most classes have misclassification into *children voices*, all except *horn*, which, with a classification accuracy of 98.48%, is also the best performing class and the only one not to have multiple misclassifications. While there were no misclassifications for *machines* under clean conditions, it is the worst performing class at 0dB SNR with a classification accuracy of just 47.78%. It also has two of the highest misclassifications into any single class, 22.22% into *glass breaking* and 15.56% into *phone rings*.

Table 5.25: Confusion matrix for test samples at 0dB SNR using CITF
(misclassifications of more than 10% have been highlighted)

	Alarms	Children Voices	Construction	Dog barking	Footsteps	Glass breaking	Gunshots	Horn	Machines	Phone rings
Alarms	86.11	7.22	0	6.67	0	0	0	0	0	0
Children voices	3.89	61.67	7.78	12.78	6.11	4.44	0	3.33	0	0
Construction	0	6.67	85.56	0	0	2.22	0	0	0	5.56
Dog barking	0	5.95	0	92.86	0	0	0	0	0	1.19
Footsteps	0	1.17	4.09	2.92	77.19	0	9.36	0.58	0	4.68
Glass breaking	0	5.00	0	0	0	91.67	0.00	0	0	3.33
Gunshots	0	2.38	11.90	0	3.57	0	82.14	0	0	0
Horn	0	0	0	1.52	0	0	0	98.48	0	0
Machines	0	11.11	3.33	0	0	22.22	0	0	47.78	15.56
Phone Rings	0	5.07	0.72	1.45	0	13.04	0	0	0	79.71
Overall Classification Accuracy = 78.30%										

To further understand the effect of the different environmental noises on the classification performance, the average classification accuracy for each noise type at 0dB SNR are computed which are as follows: *speech babble* – 69.29%, *destroyer control room* – 78.74%, and *factory floor 1* – 86.88%. This shows that most of the misclassifications are due to *speech babble* noise while *factory floor 1* has the least misclassifications. The *machines* sound class has three subclasses and, upon further analysis, it was observed that under *destroyer control room* noise, most of the test

samples from subclasses 1 and 2 were misclassified into *children voices* and *phone rings*, respectively. The cochleagram image of a sample sound signal from subclass 1 under clean conditions and with the addition of *destroyer control room* noise at 0dB SNR is shown in Figure 5.5(a) and (b), respectively. The dominant frequency components of the sound signal are clearly evident under clean conditions in Figure 5.5(a). While they are also largely visible with the addition of noise, Figure 5.5(b), the *destroyer control room* noise introduces strong spectral peaks which significantly alters the intensity distribution in the cochleagram image, hence, making the classification task much more difficult. While a decent overall classification accuracy of 78.74% is still managed to be achieved using *destroyer control room* noise at 0dB SNR, it could be said that the proposed features are more suited to noise environments which do not contain strong spectral peaks, such as *factory floor 1*, as shown in the time-frequency images in Figure 3.11.



(a) (b)

Figure 5.5: Cochleagram images of a sample sound signal from subclass 1 of sound class *machines*. (a) Cochleagram image of sound signal under clean conditions and (b) cochleagram image of sound signal at 0dB SNR with destroyer control room noise.

Moreover, compared to CITF, linear GTCCs, the best performing baseline feature, have significantly higher confusion at 0dB SNR as per the confusion matrix in Table 5.26. To some extent, there is a reversal in the classification performance of individual classes. For example, with CITF, *children voices*, *machines*, and *phone rings* are amongst the worst performing classes at 0dB SNR. However, the classification accuracy of these classes is higher with GTCCs. For all the other classes, however, CITF gives much better classification performance than GTCCs.

There are also some similar trends as far as misclassifications are concerned. With CITF, all except one class has misclassifications into *children voices* and with GTCCs, all classes have misclassifications into *children voices*. Also, misclassifications of more than 10% are most into *glass breaking* for both features, two classes for CITF and six classes for GTCCs.

Table 5.26: Confusion matrix for test samples at 0dB SNR using GTCCs (misclassifications of more than 10% have been highlighted)

	Alarms	Children Voices	Construction	Dog barking	Footsteps	Glass breaking	Gunshots	Horn	Machines	Phone rings
Alarms	48.89	15.00	0	0	0	23.89	0	0	0	12.22
Children voices	0	85.00	1.11	0.56	0	6.11	0	0	6.11	1.11
Construction	0	7.78	64.44	0	0	23.33	4.44	0	0	0
Dog barking	9.52	54.76	2.38	22.62	0	10.71	0	0	0	0
Footsteps	0	9.36	1.75	0	54.39	32.16	2.34	0	0	0
Glass breaking	0	10.00	0	0	0	81.67	0	0	3.33	5.00
Gunshots	0	7.14	15.48	0	25.00	30.95	21.43	0	0	0
Horn	1.52	25.76	1.52	0	0	1.52	0	69.70	0	0
Machines	0	3.33	0	0	1.11	18.89	0	0	61.11	15.56
Phone Rings	0	1.45	0	0	0	0.72	0	0	5.80	92.03
Overall Classification Accuracy = 61.77%										

5.8.2 Performance Analysis of the Different Classification Methods

Of the four multiclass SVM classification methods considered in this work, the OAA classification method generally gave the best overall classification performance and also the most noise robust. The better performance of the OAA classification method over the other methods under noisy conditions could be explained in terms of its decision function. In OAA method, the class corresponding to the largest margin is declared the winner indicating a high confidence level in the decision. However, in the other three multiclass SVM classification methods, the final decision is based on classification between pair of classes. The class even with the slightest of margin wins and gets a vote in the case of OAO classification

method or proceeds to the next round as in the case of DDAG and ADAG classification methods. The hyperplane between classes has been determined using clean samples only and with the addition of noise, there could be more overlapping of data points meaning the hyperplane is no longer an optimal one. As such, chances of error with the OAO, DDAG, and ADAG methods are increased more than the OAA method. The *k*NN classifier, on the other hand, gave a mixed classification performance. It was seen to be more effective for linear cepstral coefficients and linear time-frequency image features but produced poor results with log cepstral coefficients and log time-frequency image features. The performance with feature vector combination was also the worst amongst the classifiers considered. The DNN classifier, however, almost always outperformed all other classification methods both in terms of overall classification performance and noise robustness.

Next, the training and evaluation time of the four multiclass SVM classification methods and the *k*NN and DNN classifiers are compared. These are given in Table 5.27 for the best performing feature set of linear GTCC + CITF.

Table 5.27: Comparison of training and evaluation time of the different classification methods for the best performing combined feature vector (linear GTCC + CITF)

Classification Method	Training Time (s)	No. of Classifiers Evaluated per Test Sample ($M = 10$)	Total Testing Time (s)
OAA-SVM	0.4512	10	33.7504
OAO-SVM	0.4565	45	106.7339
DDAG-SVM	0.4565	9	21.6259
ADAG-SVM	0.4565	9	22.0322
*k*NN	0.0530	–	1.2498
DNN	89.8556	–	0.0566

Starting with the multiclass SVM classification methods, the OAO, DDAG, and ADAG approaches have the same training procedure and time. The training time for OAA is only marginally lower than these three classification methods. The DDAG

and ADAG classification methods have approximately the same evaluation time and are the fastest. Using the DDAG evaluation time as basis, the OAA method takes about 1.56 times more time while OAO classification method takes a significantly greater time, about 4.94 times more than DDAG. The significantly higher evaluation time for the OAO classification method can be expected since it requires the evaluation for 45 classifiers per test sample when compared to only 9 classifiers for DDAG and ADAG classification methods. As such, ideally, the OAO approach should take 5 times more time to evaluate. Since the OAA multiclass classification method generally gives more noise robust and better overall classification performance than the other three classification methods and with a reasonable training and evaluation time, it could be deduced that it is the most suitable multiclass SVM classification method from those considered.

However, the training and evaluation time of all the multiclass SVM classification methods are significantly higher than the *k*NN classifier, which also offers the fastest training time. The disadvantage of the *k*NN classifier though is the mixed classification performance for the many different features considered in this work. On the other hand, the training time of the DNN classifier is considerably higher than all classification methods, about 200 times more than OAA-SVM, which can be a disadvantage if performing unsupervised training. However, the evaluation time of the DNN classifier is the fastest, about 596 times faster than OAA-SVM. Also, the DNN classifier almost always gave the highest overall classification performance and the most noise robust. Therefore, if using supervised training, as in this work, the DNN classifier can be considered the best choice due to its superior classification performance and faster evaluation time. Besides, techniques such as the use of GPUs over CPUs have been proposed for faster training time for DNNs [168, 169].

5.8.3 Training and Evaluation Time of Features

Finally, the training and evaluation time of the different features are computed. These are plotted in Figure 5.6 for DNN classification. The training and evaluation time in this instance are largely affected by two variables, the feature vector dimension and the internal layer dimensions of the DNN classifier, both of which are given in Table 5.3. For example, all cepstral features and the RSIF and RCIF

have the same feature dimension of 72 and DNN internal layer dimension of 50. As such, the training and evaluation time of these features are approximately same. Similarly, SIF and CIF have the same feature dimension and layer dimensions resulting in approximately same training and evaluation times. In general, a good correlation is observed between the training and evaluation times.

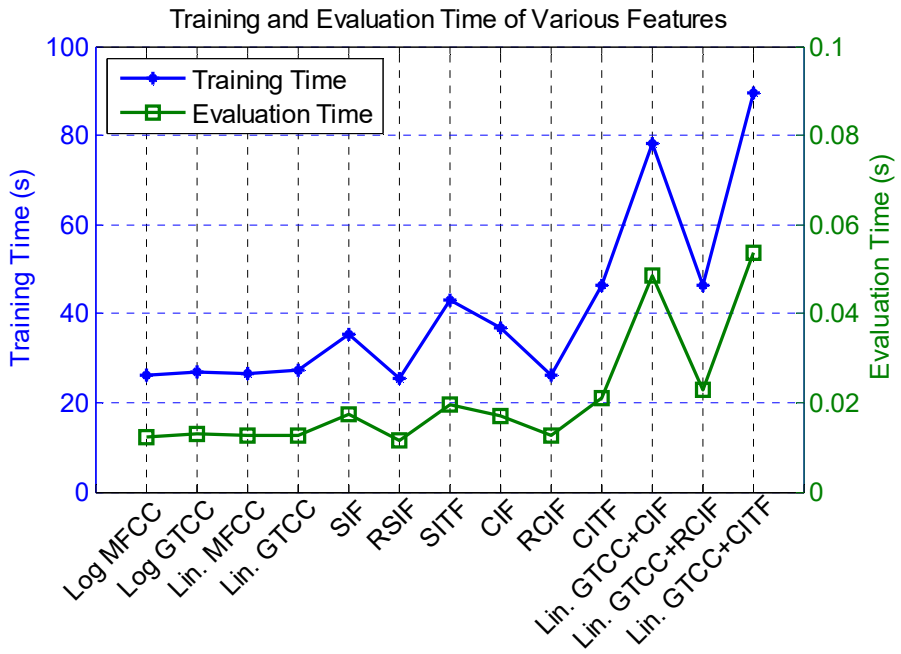


Figure 5.6: Training and evaluation time of various features

As far as the individual features are concerned, the cepstral features and the RSIF and RCIF have the fastest training and evaluation times of about 26s and 12.5ms, respectively. With a training time of more than 40s and an evaluation time of approximately 20ms, the SITF and CITF, the best performing spectrogram and cochleagram features, respectively, have the highest training and evaluation times of all the individual features. As such, the RSIF and RCIF probably offer the best compromise between classification accuracy and the training and evaluation times.

The feature combination of linear GTCC with CIF and CITF coupled with 160 dimensional internal layers results in the highest training and evaluation times. However, due to relatively lower feature and layer dimensions, the training and evaluation times of linear GTCC + RCIF is relatively low, both at about half of

linear GTCC + CITF. In addition, the average classification accuracy using this combination was determined to be 95.42%, only 0.64% lower than the feature combination of linear GTCC + CITF which gives the highest average classification performance. As such, the feature combination of linear GTCC + RCIF is a good alternative if lower computational costs are a priority.

It should be noted that the training and evaluation times given here are for indicative purposes and for relative comparison. These times were measured using software and can vary depending on the internal or background processes in the processing unit and the processing power dedicated by the processing unit, amongst others, which are not totally controllable by the user.

Chapter 6

Conclusion

This work considered a number of cepstral and time-frequency image features in trying to achieve improvement in classification performance in the presence of noise in an audio surveillance application. For cepstral coefficients, treated as baseline features in this work, using both log and root compression, GTCCs gave the best overall and most noise robust performance. Root compression was seen to significantly improve the noise robustness of both features, MFCCs and GTCCs. The root value was set to 1, around which the best overall classification performance was achieved, and this was referred as linear compression. Generally, there was a slight reduction in classification performance from log to linear compression under clean conditions but linear compression was seen to be much more effective under noisy conditions with a better overall classification accuracy.

The final baseline feature was the SIF. While the best results using the SIF were higher than log compressed MFCCs and GTCCs, it was only marginally better than linear MFCCs and lower than linear GTCCs. As such, linear GTCCs was determined as the best performing baseline feature.

The proposed reduced method for the SIF, RSIF, gave an improved classification performance when compared to the SIF with the added advantage of a much lower feature dimension. The overall classification performance was also higher than linear GTCCs. However, the best overall classification performance using spectrogram derived features was achieved with the SITF, which is based on the GLCM method of image texture analysis. While the classification accuracy under clean conditions was slightly lower than linear GTCCs, the classification

performance under noisy conditions, particularly at low SNRs, was significantly better. Also, for all three time-frequency image features, significantly improved classification performance was achieved using cochleagram image feature extraction instead of the conventional spectrogram image. These features were referred as CIF, RCIF, and CITF and the CITF gave the best overall performance, as did SITF with spectrogram feature extraction.

As far as feature combination is concerned, only GTCCs were considered being the best performing cepstral feature. Of the time and frequency domain features considered in this work, the inclusion of only some features were seen to have a positive effect on the classification performance with SBE determined to be the most effective. However, the combination with cochleagram image derived features was seen to give better classification performance and the combination with CITF determined to give the highest results.

In addition, of the four multiclass SVM classification methods considered in this work, the classification performance of the OAA method was generally seen to be the best with both individual and combined features. However, the overall classification performance of the DNN classifier was the highest from all the classifiers considered with all but one feature. The DNN classifier also had a much faster evaluation time but the training time was determined to be the slowest. This can be an issue in an unsupervised system or when performing training in real-time. Apart from the number of layers and layer dimension, the training and evaluation time are also dependent on the feature dimension. As a tradeoff between classification accuracy and the training and evaluation time, the feature combination of linear GTCC and RCIF was determined to be arguably the best.

While the proposed features show improvement in classification performance when compared to related work, there are still a number of areas to improve on. The proposed time-frequency image features were determined to be more suited to noise types which do not contain strong spectral peaks. As such, more research is needed to test and improve the performance in the presence of impulsive noise. In addition, this work did not consider out-of-class sound signals which would be beneficial for a practical implementation of an audio surveillance system. For the sound database used, the sound signals are either already segmented or segmentation is performed manually. As such, another requirement for a real-time implementation is sound

signal segmentation. In [27], for example, the sum of the signal magnitude is used to distinguish between silent and non-silent frames. Separation of overlapping sound events will also be required for a more robust performance, such as in [170].

In addition, the inconsistency in the choice of sound databases in most literature makes it difficult to make direct comparison of the performance of the proposed techniques. While sound libraries, such as the Latin music database, RWCP databases, and the BBC sound effects library have been employed for research in certain SER applications, the creation of the sound database for use from these available libraries is at the discretion of the researchers. Also, the number and complexity of sound classes and the amount of training data, amongst others, have a direct influence on the classification performance of a SER system. Therefore, there is a need to standardize sound databases and experimental setups to make it easier for direct comparison of proposed techniques, similar to what has been seen in [26-28], refer to Table 2.1.

Moreover, different approaches have been noticed in structuring of classes in some similar applications, such as audio surveillance, as in [5], and sound event recognition, as in [2]. In [5], a sound class has a number of sound events. For example, shots fired from a rifle, shotgun, and machine gun are examples of different sound events but would be treated as a single sound class such as gunshots. In some cases, the signal properties of subclasses in a particular class are similar to the subclasses in other classes but different to subclasses in its own class. This creates interclass similarity and intraclass diversity, increasing the complexity of the problem as a result.

Furthermore, the addition of new features does not necessarily improve the classification performance as some features are redundant, as seen in 5.7.1, and optimization techniques have been used in some literature to determine the optimal feature set. Alexandre et al. [32] argue the computational limitations of digital signal processing hardware in hearing aids and genetic algorithm (GA) with restricted search [171] is proposed to select the optimal features so that the feature vector dimension could be reduced and the computation speed increased as a result. With an original 76-dimensional feature vector, the results show that while the unconstrained GA required 43 and 46 features to get the best probability of correct classification for the two classification problems, respectively, only 11 features are

shown to give comparable performance using restricted GA which is also always slightly better than the sequential methods [172], sequential forward search (SFS) and sequential backward search (SBS). Chmulik and Jarina [173] experimented with particle swarm optimization (PSO) [174] and GA to select the optimum features for classification of six sound classes. While comparable classification accuracy is achieved using both the optimization techniques, PSO gives the highest classification accuracy at 82.48% with a feature dimension of 86. This is much better than the classification accuracy with all the features included at 72.94% and a feature dimension of 137.

References

[1] D. Mitrović, M. Zeppelzauer, and C. Breiteneder, "Features for Content-Based Audio Retrieval," in *Advances in Computers*. vol. 78, V. Z. Marvin, Ed. Elsevier, 2010, pp. 71-150.

[2] J. Dennis, H. D. Tran, and H. Li, "Spectrogram image feature for sound event classification in mismatched conditions," *IEEE Signal Processing Letters,* vol. 18, no. 2, pp. 130-133, 2011.

[3] K. K. Paliwal, "Spectral subband centroid features for speech recognition," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, 1998, pp. 617-620.

[4] B. Gajic and K. K. Paliwal, "Robust speech recognition in noisy environments based on subband spectral centroid histograms," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 14, no. 2, pp. 600-608, 2006.

[5] A. Rabaoui, M. Davy, S. Rossignol, and N. Ellouze, "Using one-class SVMs and wavelets for audio surveillance," *IEEE Transactions on Information Forensics and Security,* vol. 3, no. 4, pp. 763-775, 2008.

[6] J. Kotus, K. Lopatka, A. Czyżewski, and G. Bogdanis, "Audio-visual surveillance system for application in bank operating room," in *Multimedia Communications, Services and Security*. vol. 368, A. Dziech and A. Czyżewski, Eds. Springer Berlin Heidelberg, 2013, pp. 107-120.

[7] B. Uzkent, B. D. Barkana, and H. Cevikalp, "Non-speech environmental sound classification using SVMs with a new set of features," *International Journal of Innovative Computing, Information and Control,* vol. 8, no. 5(B), pp. 3511-3524, 2012.

[8] W. Huang, S. Lau, T. Tan, L. Li, and L. Wyse, "Audio events classification using hierarchical structure," in *Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia*, Singapore, 2003, pp. 1299-1303.

[9] S. Chu, S. Narayanan, and C. C. J. Kuo, "Environmental sound recognition with time-frequency audio features," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 17, no. 6, pp. 1142-1158, 2009.

[10] X. Valero and F. Alias, "Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification," *IEEE Transactions on Multimedia,* vol. 14, no. 6, pp. 1684-1689, 2012.

[11] O. Cheng, W. Abdulla, and Z. Salcic, "Performance evaluation of front-end processing for speech recognition systems," The University of Auckland, New Zealand, Report 621, 2005.

[12] P. Alexandre and P. Lockwood, "Root cepstral analysis: A unified view. Application to speech processing in car noise environments," *Speech Communication,* vol. 12, no. 3, pp. 277-288, 1993.

[13] R. Sarikaya and J. H. Hansen, "Analysis of the root-cepstrum for acoustic modeling and fast decoding in speech recognition," in *EUROSPEECH-2001*, Aalborg, Denmark, 2001, pp. 687-690.

[14] R. V. Sharan and T. J. Moir, "An overview of applications and advancements in automatic sound recognition," *Neurocomputing,* vol. 200, pp. 22-34, 2016.

[15] R. V. Sharan and T. J. Moir, "Audio surveillance under noisy conditions using time-frequency image feature," in *Proceedings of the 19th International Conference on Digital Signal Processing (DSP 2014)*, Hong Kong, 2014, pp. 130-135.

[16] R. V. Sharan and T. J. Moir, "Noise robust audio surveillance using reduced spectrogram image feature and one-against-all SVM," *Neurocomputing,* vol. 158, pp. 90-99, 2015.

[17] R. V. Sharan and T. J. Moir, "Robust audio surveillance using spectrogram image texture feature," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia, 2015, pp. 1956-1960.

[18] R. V. Sharan and T. J. Moir, "Subband time-frequency image texture features for robust audio surveillance," *IEEE Transactions on Information Forensics and Security,* vol. 10, no. 12, pp. 2605-2615, 2015.

[19] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics,* vol. SMC-3, no. 6, pp. 610-621, 1973.

[20] A. Eleyan and H. Demirel, "Co-occurrence matrix and its statistical features as a new approach for face recognition," *Turkish Journal of Electrical Engineering & Computer Sciences,* vol. 19, no. 1, pp. 97-107, 2011.

[21] Y. M. G. Costa, L. S. Oliveira, A. L. Koericb, and F. Gouyon, "Music genre recognition using spectrograms," in *18th International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2011, pp. 1-4.

[22] R. V. Sharan and T. J. Moir, "Cochleagram image feature for improved robustness in sound recognition," in *Proceedings of the IEEE International Conference on Digital Signal Processing (DSP)*, Singapore, 2015, pp. 441-444.

[23] D. Wang and G. J. Brown, "Fundamentals of computational auditory scene analysis," in *Computational auditory scene analysis: Principles, algorithms and applications*, D. Wang and G. J. Brown, Eds. IEEE Press/Wiley-Interscience, 2006, pp. 1-44.

[24] R. V. Sharan and T. J. Moir, "Comparison of multiclass SVM classification techniques in an audio surveillance application under mismatched conditions," in *Proceedings of the 19th International Conference on Digital Signal Processing (DSP 2014)*, Hong Kong, 2014, pp. 83-88.

[25] R. V. Sharan and T. J. Moir, "Subband spectral histogram feature for improved sound recognition in low SNR conditions," in *Proceedings of the IEEE International Conference on Digital Signal Processing (DSP)*, Singapore, 2015, pp. 432-435.

[26] E. Wold, T. Blum, D. Keislar, and J. Wheaten, "Content-based classification, search, and retrieval of audio," *IEEE MultiMedia,* vol. 3, no. 3, pp. 27-36, 1996.

[27] S. Z. Li, "Content-based audio classification and retrieval using the nearest feature line method," *IEEE Transactions on Speech and Audio Processing,* vol. 8, no. 5, pp. 619-625, 2000.

[28] G. Guo and S. Z. Li, "Content-based audio classification and retrieval by support vector machines," *IEEE Transactions on Neural Networks,* vol. 14, no. 1, pp. 209-215, 2003.

[29] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing,* vol. 10, no. 5, pp. 293-302, 2002.

[30] A. A. Wieczorkowska, Z. W. Ras, Z. Xin, and R. Lewis, "Multi-way hierarchic classification of musical instrument sounds," in *International*

Conference on Multimedia and Ubiquitous Engineering (MUE '07), 2007, pp. 897-902.

[31] L. Lu, H.-J. Zhang, and H. Jiang, "Content analysis for audio classification and segmentation," *IEEE Transactions on Speech and Audio Processing,* vol. 10, no. 7, pp. 504-516, 2002.

[32] E. Alexandre, L. Cuadra, M. Rosa, and F. Lopez-Ferreras, "Feature selection for sound classification in hearing aids through restricted search driven by genetic algorithms," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 15, no. 8, pp. 2249-2256, 2007.

[33] A. R. Abu-El-Quran, R. A. Goubran, and A. D. C. Chan, "Security monitoring using microphone arrays and audio classification," *IEEE Transactions on Instrumentation and Measurement,* vol. 55, no. 4, pp. 1025-1032, 2006.

[34] J. L. Rouas, J. Louradour, and S. Ambellouis, "Audio events detection in public transport vehicle," in *IEEE Intelligent Transportation Systems Conference (ITSC '06)*, 2006, pp. 733-738.

[35] R. Radhakrishnan, A. Divakaran, and P. Smaragdis, "Audio analysis for surveillance applications," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005, pp. 158-161.

[36] M. V. Ghiurcau, C. Rusu, R. C. Bilcu, and J. Astola, "Audio based solutions for detecting intruders in wild areas," *Signal Processing,* vol. 92, no. 3, pp. 829-840, 2012.

[37] D. Istrate, E. Castelli, M. Vacher, L. Besacier, and J. F. Serignat, "Information extraction from sound for medical telemonitoring," *IEEE Transactions on Information Technology in Biomedicine,* vol. 10, no. 2, pp. 264-274, 2006.

[38] *Muscle Fish*. Available: http://www.musclefish.com

[39] L. Lu, H.-J. Zhang, and S. Z. Li, "Content-based audio classification and segmentation by using support vector machines," *Multimedia Systems,* vol. 8, no. 6, pp. 482-492, 2003.

[40] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE,* vol. 63, no. 4, pp. 561-580, 1975.

[41] L. R. Rabiner and R. W. Schafer, *Digital processing of speech signals.* New Jersey: Prentice Hall, 1978.

[42] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 28, no. 4, pp. 357-366, 1980.

[43] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The Journal of the Acoustical Society of America,* vol. 8, no. 3, pp. 185-190, 1937.

[44] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu*, et al.*, *The HTK book (for HTK version 3.4).* Cambridge University: Engineering Department, 2009.

[45] B. Lei, S. A. Rahman, and I. Song, "Content-based classification of breath sound with enhanced features," *Neurocomputing,* vol. 141, pp. 139-147, 2014.

[46] X. Zhang and Y. Li, "Environmental sound recognition using double-level energy detection," *Journal of Signal and Information Processing,* vol. 4, no. 3B, pp. 19-24, 2013.

[47] B. Gao and W. L. Woo, "Wearable audio monitoring: Content-based processing methodology and implementation," *IEEE Transactions on Human-Machine Systems,* vol. 44, no. 2, pp. 222-233, 2014.

[48] C. Kim and R. M. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4101-4104.

[49] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks,* vol. 10, no. 3, pp. 626-634, 1999.

[50] J.-C. Wang, H.-P. Lee, J.-F. Wang, and C.-B. Lin, "Robust environmental sound recognition for home automation," *IEEE Transactions on Automation Science and Engineering,* vol. 5, no. 1, pp. 25-31, 2008.

[51] R. D. Patterson, K. Robinson, J. Holdsworth, D. McKeown, C. Zhang, and M. Allerhand, "Complex sounds and auditory images," in *Auditory physiology and perception*. vol. 83, Y. Cazals, L. Demany, and K. Horner, Eds. Pergamon, Oxford, 1992, pp. 429-446.

[52] M. Slaney, "An efficient implementation of the Patterson-Holdsworth auditory filter bank," Apple Computer, Technical Report 35, 1993.

[53] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing,* vol. 20, no. 1, pp. 33-61, 1998.

[54] *BBC Sound Effects Library*. Available: http://www.leonardosoft.com

[55] *The Freesound Project*. Available: http://freesound.iua.upf.edu/index.php

[56] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing,* vol. 41, no. 12, pp. 3397-3415, 1993.

[57] P. Vera-Candeas, N. Ruiz-Reyes, M. Rosa-Zurera, D. Martinez-Munoz, and F. Lopez-Ferreras, "Transient modeling by matching pursuits with a wavelet dictionary for parametric audio coding," *IEEE Signal Processing Letters,* vol. 11, no. 3, pp. 349-352, 2004.

[58] G. Yang, Q. Zhang, and P.-W. Que, "Matching-pursuit-based adaptive wavelet-packet atomic decomposition applied in ultrasonic inspection," *Russian Journal of Nondestructive Testing,* vol. 43, no. 1, pp. 62-68, 2007.

[59] R. Gribonval, "Fast matching pursuit with a multiscale dictionary of Gaussian chirps," *IEEE Transactions on Signal Processing,* vol. 49, no. 5, pp. 994-1001, 2001.

[60] S. Ghofrani, D. C. McLernon, and A. Ayatollahi, "Comparing Gaussian and chirplet dictionaries for time-frequency analysis using matching pursuit decomposition," in *Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology*, 2003, pp. 713-716.

[61] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa*, et al.*, "Audio-based context recognition," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 14, no. 1, pp. 321-329, 2006.

[62] R. Gribonval and E. Bacry, "Harmonic decomposition of audio signals with matching pursuit," *IEEE Transactions on Signal Processing,* vol. 51, no. 1, pp. 101-111, 2003.

[63] K. Umapathy, S. Krishnan, and S. Jimaa, "Multigroup classification of audio signals using time-frequency parameters," *IEEE Transactions on Multimedia,* vol. 7, no. 2, pp. 308-315, 2005.

[64] S. P. Ebenezer, A. Papandreou-Suppappola, and S. B. Suppappola, "Classification of acoustic emissions using modified matching pursuit," *EURASIP Journal on Advances in Signal Processing,* vol. 2004, no. 3, pp. 347-357, 2004.

[65] E. Smith and M. S. Lewicki, "Efficient coding of time-relative structure using spikes," *Neural Computation,* vol. 17, no. 1, pp. 19-45, 2005.

[66] S. Scholler and H. Purwins, "Sparse approximations for drum sound classification," *IEEE Journal of Selected Topics in Signal Processing,* vol. 5, no. 5, pp. 933-940, 2011.

[67] O. Gillet and G. Richard, "ENST-Drums: An extensive audio-visual database for drum signals processing," in *Proceedings of 7th International Conference on Music Information Retrieval*, Victoria, Canada, 2006, pp. 156-159.

[68] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Music genre database and musical instrument sound database," in *Proceedings of the 4th International Conference on Music Information Retrieval*, Baltimore, Maryland, USA, 2003, pp. 229–230.

[69] K. Abe, H. Sakaue, T. Okuno, and K. Terada, "Sound classification for hearing aids using time-frequency images," in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PacRim)*, 2011, pp. 719-724.

[70] S. Nakamura, K. Hiyane, F. Asano, T. Nishiura, and T. Yamada, "Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition," in *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC 2000)*, Athens, Greece, 2000, pp. 965–968.

[71] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication,* vol. 12, no. 3, pp. 247-251, Jul. 1993.

[72] *HTK Toolkit*. Available: http://htk.eng.cam.ac.uk

[73] C. N. Silla Jr., A. L. Koerich, and C. A. A. Kaestner, "The Latin music database," in *Proceedings of the 9th International Conference on Music Information Retrieval*, Philadelphia, PA, USA, 2008, pp. 451-456.

[74] M. Lopes, F. Gouyon, A. L. Koerich, and L. E. S. Oliveira, "Selection of training instances for music genre classification," in *20th International Conference on Pattern Recognition (ICPR)*, 2010, pp. 4569-4572.

[75] L.-Q. Zhu and Z. Zhang, "Auto-classification of insect images based on color histogram and GLCM," in *Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2010, pp. 2589-2593.

[76] X. Wang and N. D. Georganas, "GLCM texture based fractal method for evaluating fabric surface roughness," in *Canadian Conference on Electrical and Computer Engineering (CCECE '09)*, 2009, pp. 104-107.

[77] M. Umaselvi, S. S. Kumar, and M. Athithya, "Color based urban and agricultural land classification by GLCM texture features," in *IET Chennai 3rd International Conference on Sustainable Energy and Intelligent Systems (SEISCON 2012)*, 2012, pp. 1-4.

[78] D. Mitrea, M. Socaciu, R. Badea, and A. Golea, "Texture based characterization and automatic diagnosis of the abdominal tumors from ultrasound images using third order GLCM features," in *4th International Congress on Image and Signal Processing (CISP)*, Shanghai, 2011, pp. 1558-1562.

[79] S. Beura, B. Majhi, and R. Dash, "Mammogram classification using two dimensional discrete wavelet transform and gray-level co-occurrence matrix for detection of breast cancer," *Neurocomputing,* vol. 154, pp. 1-14, 2015.

[80] S. Mallat, *A wavelet tour of signal processing.* New York: Academic Press, 1999.

[81] S. Nilufar, N. Ray, M. K. I. Molla, and K. Hirose, "Spectrogram based features selection using multiple kernel learning for speech/music discrimination," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 501-504.

[82] S. Arivazhagan and L. Ganesan, "Texture classification using wavelet transform," *Pattern Recognition Letters,* vol. 24, no. 9–10, pp. 1513-1521, June 2003.

[83] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research,* vol. 5, pp. 27-72, 2004.

[84] E. W. Healy, S. E. Yoho, Y. Wang, and D. Wang, "An algorithm to improve speech recognition in noise for hearing-impaired listeners," *The Journal of the Acoustical Society of America,* vol. 134, no. 4, pp. 3029-3038, 2013.

[85] X.-L. Zhang and D. Wang, "Boosted deep neural networks and multi-resolution cochleagram features for voice activity detection," The Ohio State University, Columbus, OH, Technical Report OSU-CISRC-4/14-TR0, 2014.

[86] B. Gao, W. L. Woo, and L. C. Khor, "Cochleagram-based audio pattern separation using two-dimensional non-negative matrix factorization with automatic sparsity adaptation," *The Journal of the Acoustical Society of America,* vol. 135, no. 3, pp. 1171-85, Mar 2014.

[87] V. Vapnik and A. Lerner, "Pattern recognition using generalized portrait method," *Automation and Remote Control,* vol. 24, no. 6, pp. 774-780, 1963.

[88] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, Pittsburgh, Pennsylvania, USA, 1992, pp. 144-152.

[89] V. N. Vapnik, *Statistical learning theory.* New York: Wiley, 1998.

[90] U. H. G. Kreßel, "Pairwise classification and support vector machines," in *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1999, pp. 255-268.

[91] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification," in *Advances in Neural Information Processing Systems 12 (NIPS-99)*, S. A. Solla, T. K. Leen, and K.-R. Müller, Eds. Cambridge MA: MIT Press, 2000, pp. 547-553.

[92] B. Kijsirikul, N. Ussivakul, and S. Meknavin, "Adaptive directed acyclic graphs for multiclass classification," in *PRICAI 2002: Trends in Artificial Intelligence*. vol. 2417, M. Ishizuka and A. Sattar, Eds. Berlin Heidelberg: Springer, 2002, pp. 158-168.

[93] C.-C. Chang and C.-J. Lin, "LIBSVM : A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology,* vol. 2, no. 3, pp. 1-27, 2011.

[94] S. Kolozali, M. Barthet, G. Fazekas, and M. Sandler, "Automatic ontology generation for musical instruments based on audio analysis," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 21, no. 10, pp. 2207-2220, 2013.

[95] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks,* vol. 5, no. 6, pp. 989-993, 1994.

[96] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks,* vol. 13, no. 2, pp. 415-425, 2002.

[97] N. Seo, "A comparison of multi-class support vector machine methods for face recognition," The University of Maryland, Research Report, 6 Dec 2007.

[98] J. Weston and C. Watkins, "Multi-class support vector machines," Department of Computer Science, Royal Holloway, University of London, Egham, UK, Technical Report CSD-TR-98-04, 1998.

[99] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of Machine Learning Research,* vol. 2, pp. 265-292, 2001.

[100] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," Microsoft Research, Technical Report MSR-TR-99-87, 1999.

[101] *Leonardo Software*. Available: http://www.leonardosoft.com

[102] G. Valenzise, L. Gerosa, M. Tagliasacchi, E. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS 2007)*, 2007, pp. 21-26.

[103] P. K. Atrey, M. C. Maddage, and M. S. Kankanhalli, "Audio based event detection for multimedia surveillance," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06)*, 2006, pp. 813-816.

[104] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation,* vol. 18, no. 7, pp. 1527-1554, 2006.

[105] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly*, et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine,* vol. 29, no. 6, pp. 82-97, 2012.

[106] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, and W. Xiao, "Robust Sound Event Classification Using Deep Neural Networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 23, no. 3, pp. 540-552, 2015.

[107] T. C. Walters, "Auditory-based processing of communication sounds," Ph.D. dissertation, University of Cambridge, Cambridge, U.K., 2011.

[108] S. Chu, S. Narayanan, and C. C. J. Kuo, "Environmental sound recognition using MP-based features," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*, 2008, pp. 1-4.

[109] J. Dennis, "Sound event recognition in unstructured environments using spectrogram image processing," Ph.D. Dissertation, Nanyang Technological University, Singapore, 2014.

[110] F. Beritelli and A. Spadaccini, "Human identity verification based on Mel frequency analysis of digital heart sounds," in *16th International Conference on Digital Signal Processing*, 2009, pp. 1-5.

[111] C. Kwak and O. W. Kwon, "Cardiac disorder classification by heart sound signals using murmur likelihood and hidden markov model state likelihood," *IET Signal Processing,* vol. 6, no. 4, pp. 326-334, 2012.

[112] G.-C. Chang and Y.-P. Cheng, "Investigation of noise effect on lung sound recognition," in *International Conference on Machine Learning and Cybernetics*, 2008, pp. 1298-1301.

[113] *Stethographics*. Available: http://www.stethographics.com

[114] S. M. Kay, *Modern spectral estimation: Theory and application.* Englewood Cliffs, NJ: Prentice Hall, 1988.

[115] Y. Liu, C.-M. Zhang, F. Feng, and S.-J. Li, "Lung sound feature extraction based on parametric bispectrum analysis of higher-order cumulants " *Journal of Shandong University (Engineering Science),* vol. 35, no. 2, pp. 77-85, 2005.

[116] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 26, no. 1, pp. 43-49, 1978.

[117] A. Azarbarzin and Z. M. K. Moussavi, "Automatic and unsupervised snore sound extraction from respiratory sound signals," *IEEE Transactions on Biomedical Engineering,* vol. 58, no. 5, pp. 1156-1162, 2011.

[118] V. Exadaktylos, M. Silva, J. M. Aerts, C. J. Taylor, and D. Berckmans, "Real-time recognition of sick pig cough sounds," *Computers and Electronics in Agriculture,* vol. 63, no. 2, pp. 207-214, 2008.

[119] S. Ferrari, R. Piccinini, M. Silva, V. Exadaktylos, D. Berckmans, and M. Guarino, "Cough sound description in relation to respiratory diseases in dairy calves," *Preventive Veterinary Medicine,* vol. 96, no. 3–4, pp. 276-280, 2010.

[120] C. Dimoulas, G. Kalliris, G. Papanikolaou, V. Petridis, and A. Kalampakas, "Bowel-sound pattern analysis using wavelets and neural networks with application to long-term, unsupervised, gastrointestinal motility monitoring," *Expert Systems with Applications,* vol. 34, no. 1, pp. 26-41, 2008.

[121] J. Cai, D. Ee, B. Pham, P. Roe, and J. Zhang, "Sensor network for the monitoring of ecosystem: Bird species recognition," in *3rd International*

Conference on Intelligent Sensors, Sensor Networks and Information, 2007, pp. 293-298.

[122] H. Jaafar and D. A. Ramli, "Automatic syllables segmentation for frog identification system," in *2013 IEEE 9th International Colloquium on Signal Processing and its Applications (CSPA)*, 2013, pp. 224-228.

[123] J. C. Brown and P. Smaragdis, "Hidden Markov and Gaussian mixture models for automatic call classification," *Journal of the Acoustical Society of America,* vol. 125, no. 6, pp. EL221-EL224, 2009.

[124] M. Brookes. VOICEBOX: Speech processing toolbox for MATLAB [Online]. Available: http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

[125] Z. Le-Qing, "Insect sound recognition based on MFCC and PNN," in *2011 International Conference on Multimedia and Signal Processing (CMSP)*, 2011, pp. 42-46.

[126] D. H. Milone, J. R. Galli, C. A. Cangiano, H. L. Rufiner, and E. A. Laca, "Automatic recognition of ingestive sounds of cattle based on hidden Markov models," *Computers and Electronics in Agriculture,* vol. 87, pp. 51-55, 2012.

[127] A. Aydin, C. Bahr, S. Viazzi, V. Exadaktylos, J. Buyse, and D. Berckmans, "A novel method to automatically measure the feed intake of broiler chickens by sound technology," *Computers and Electronics in Agriculture,* vol. 101, pp. 17-23, 2014.

[128] D. V. Smith and M. S. Shahriar, "A context aware sound classifier applied to prawn feed monitoring and energy disaggregation," *Knowledge-Based Systems,* vol. 52, pp. 21-31, 2013.

[129] V. T. Vu, F. Bremond, G. Davini, M. Thonnat, P. Quoc-Cuong, N. Allezard, *et al.*, "Audio-video event recognition system for public transport security," in *The Institution of Engineering and Technology Conference on Crime and Security, 2006*, 2006, pp. 414-419.

[130] J.-X. Du, C.-M. Zhai, Y.-L. Guo, Y.-Y. Tang, and C. L. P. Chen, "Recognizing complex events in real movies by combining audio and video features," *Neurocomputing,* vol. 137, pp. 89-95, 2014.

[131] S. Chu, S. Narayanan, C. C. J. Kuo, and M. J. Mataric, "Where am I? Scene recognition for mobile robots using audio features," in *2006 IEEE International Conference on Multimedia and Expo*, 2006, pp. 885-888.

[132] E. Menegatti, M. Cavasin, E. Pagello, E. Mumolo, and M. Nolich, "Combining audio and video surveillance with a mobile robot," *International Journal on Artificial Intelligence Tools,* vol. 16, no. 2, pp. 377-398, 2007.

[133] Z. Cheng, X. Zhang, S. Yu, Y. Ou, X. Wu, and Y. Xu, "A surveillance robot with human recognition based on video and audio," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2010, pp. 1256-1261.

[134] Q. Zhang, F.-Q. Zhao, Z.-J. Liu, and P. Yang, "Audio sensors fusion based on vote for robot navigation," in *25th Chinese Control and Decision Conference (CCDC)*, 2013, pp. 3219-3222.

[135] Y. Yao, G. Bin, Y. Zhiwen, and H. Huilei, "Social activity recognition and recommendation based on mobile sound sensing," in *2013 IEEE 10th International Conference on Ubiquitous Intelligence & Computing and 2013 IEEE 10th International Conference on Autonomic & Trusted Computing (UIC/ATC)*, 2013, pp. 103-110.

[136] C. Woo-Hyun, K. Seung-Il, K. Min-Seok, D. K. Han, and K. Hanseok, "Acoustic and visual signal based context awareness system for mobile application," *IEEE Transactions on Consumer Electronics,* vol. 57, no. 2, pp. 738-746, 2011.

[137] F. Tong, X.-M. Xu, S. K. Tso, and K. P. Liu, "Application of evolutionary neural network in impact acoustics based nondestructive inspection of tile-wall," in *Proceedings of International Conference on Communications, Circuits and Systems*, 2005, pp. 974-978.

[138] M. Márquez-Molina, L. P. Sánchez-Fernández, S. Suárez-Guerra, and L. A. Sánchez-Pérez, "Aircraft take-off noises classification based on human auditory's matched features extraction," *Applied Acoustics,* vol. 84, pp. 83-90, Oct. 2014.

[139] G. A. Montazer, R. Sabzevari, and H. G. Khatir, "Improvement of learning algorithms for RBF neural networks in a helicopter sound identification system," *Neurocomputing,* vol. 71, no. 1–3, pp. 167-173, 2007.

[140] M. D. Redel-Macías, F. Fernández-Navarro, P. A. Gutiérrez, A. J. Cubero-Atienza, and C. Hervás-Martínez, "Ensembles of evolutionary product unit or RBF neural networks for the identification of sound for pass-by noise test in vehicles," *Neurocomputing,* vol. 109, pp. 56-65, 2013.

[141] J. Lee and A. Rakotonirainy, "Acoustic hazard detection for pedestrians with obscured hearing," *IEEE Transactions on Intelligent Transportation Systems,* vol. 12, no. 4, pp. 1640-1649, 2011.

[142] M. Tabacchi, C. Asensio, I. Pavón, M. Recuero, J. Mir, and M. C. Artal, "A statistical pattern recognition approach for the classification of cooking stages. The boiling water case," *Applied Acoustics,* vol. 74, no. 8, pp. 1022-1032, 2013.

[143] D. Mason, *Listening to the heart: A comprehensive collection of heart sounds and murmurs.* 2nd ed. Philadelphia: F. A. Davis Company, 2000.

[144] F. V. Gessel. Top 40 Bird Songs [Online]. Available: http://www.birdsinbackyards.net

[145] D. Stewart, "Australian bird calls: Subtropical east," *CD, Nature Sound,* 2002.

[146] D. Stewart, "Voices of subtropical rainforests," *CD, Nature Sound,* 2002.

[147] R. Mankin. *Sound Library*. Available: http://www.ars.usda.gov

[148] D. F. Specht, "Probabilistic neural networks," *Neural Networks,* vol. 3, pp. 109-118, 1990.

[149] "IEC 1260: Electroacoustics - Octave-band and fractional-octave-band filters," *International Electrotech Commission,* 1995.

[150] "ANSI Standard S1.11-2004: Specification for octave-band and fractional-octave-band analog and digital filters," *American National Standards Institute,* 2004.

[151] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics,* vol. 33, no. 3, pp. 1065-1076, 1962.

[152] D. O'Shaughnessy, *Speech communication: human and machine.* Addison-Wesley Pub. Co., 1987.

[153] B. R. Glasberg and B. C. Moore, "Derivation of auditory filter shapes from notched-noise data," *Hearing Research,* vol. 47, no. 1-2, pp. 103-138, 1990.

[154] M. Slaney, "Lyon's Cochlear Model," Apple Computer, Technical Report 13, 1988.

[155] D. D. Greenwood, "A cochlear frequency-position function for several species - 29 years later," *Journal of the Acoustical Society of America* vol. 87, no. 6, pp. 2592-2605, Jun 1990.

[156] M. Slaney, "Auditory Toolbox for Matlab," Interval Research Corproation, Technical Report 1998-010, 1998.

[157] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning,* vol. 20, no. 3, pp. 273-297, Sep. 1995.

[158] G. Madzarov and D. Gjorgjevikj, "Evaluation of distance measures for multi-class classification in binary SVM decision tree," in *Artificial Intelligence and Soft Computing*. vol. 6113, L. Rutkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh, and J. Zurada, Eds. Berlin Heidelberg: Springer, 2010, pp. 437-444.

[159] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B (Methodological),* vol. 39, no. 1, pp. 1-38, 1977.

[160] R. B. Palm, "Prediction as a candidate for learning deep hierarchical models of data," M.Sc. Thesis, Technical University of Denmark, Lyngby, Denmark, 2012.

[161] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in *Parallel distributed processing*. vol. 1, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge: MIT Press, 1986, pp. 194-281.

[162] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation,* vol. 14, no. 8, pp. 1771-1800, 2002.

[163] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature,* vol. 323, no. 6088, pp. 533-536, 1986.

[164] J.-C. Wang, J.-F. Wang, K. W. He, and C.-S. Hsu, "Environmental sound classification using hybrid SVM/KNN classifier and MPEG-7 audio low-level descriptor," in *International Joint Conference on Neural Networks (IJCNN '06)*, 2006, pp. 1731-1735.

[165] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory,* vol. 28, no. 2, pp. 129-137, 1982.

[166] S. Ravindran, D. V. Anderson, and M. Slaney, "Improving the noise robustness of mel-frequency cepstral coefficients for speech processing," in *Proceedings of the ISCA Workshop on Statistical and Perceptual Audition*, Pittsburgh, PA, 2006, pp. 48-52.

[167] V. Tyagi and C. Wellekens, "On desensitizing the mel-cepstrum to spurious spectral components for robust speech recognition," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, 2005, pp. 529-532.

[168] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, *et al.*, "Theano: A CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, Austin, TX, 2010, pp. 1-3.

[169] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale deep unsupervised learning using graphics processors," in *Proceedings of the 26th Annual International Conference on Machine Learning*, Montreal, Quebec, Canada, 2009, pp. 873-880.

[170] F. Briggs, B. Lakshminarayanan, L. Neal, X. Z. Fern, R. Raich, S. J. Hadley, *et al.*, "Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach," *Journal of the Acoustical Society of America,* vol. 131, no. 6, pp. 4640-4650, 2012.

[171] S. Salcedo-Sanz, G. Camps-Valls, F. Perez-Cruz, J. Sepulveda-Sanchis, and C. Bousono-Calzon, "Enhancing genetic feature selection through restricted search and Walsh analysis," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* vol. 34, no. 4, pp. 398-406, 2004.

[172] C. M. Bishop, *Neural networks for pattern recognition.* Oxford: Clarendon Press, 1995.

[173] M. Chmulik and R. Jarina, "Bio-inspired optimization of acoustic features for generic sound recognition," in *19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2012, pp. 629-632.

[174] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, 1995, pp. 1942-1948.