ACOUSTIC SCENE CLASSIFICATION USING SPEECH FEATURES

Thesis

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

MANJUNATH MULIMANI

(155098 CS15FV06)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,

SURATHKAL, MANGALORE - 575025

NOVEMBER 2020

Dedicated to
my loving family, teachers and friends

DECLARATION

by the Ph.D. Research Scholar

I hereby declare that the Research Thesis entitled **ACOUSTIC SCENE CLASSIFICATION USING SPEECH FEATURES** which is being submitted to the **National Institute of Technology Karnataka, Surathkal** in partial fulfilment of the requirements for the award of the Degree of **Doctor of Philosophy** in **Computer Science and Engineering** is a *bonafide report of the research work carried out by me.* The material contained in this Research Thesis has not been submitted to any University or Institution for the award of any degree.

(155098 CS15FV06, Manjunath Mulimani)
Department of Computer Science and Engineering

Place: NITK, Surathkal.
Date:

CERTIFICATE

This is to *certify* that the Research Thesis entitled **ACOUSTIC SCENE CLASSIFICATION USING SPEECH FEATURES** submitted by **MANJUNATH MULIMANI**, (Register Number: 155098 CS15FV06) as the record of the research work carried out by him, is *accepted as the Research Thesis submission* in partial fulfilment of the requirements for the award of degree of **Doctor of Philosophy**.

Dr. Shashidhar G. Koolagudi
Research Guide

Dr. Alwyn Roshan Pais
Chairman - DRPC

Acknowledgment

Writing the note of gratitude is the finishing touch on my dissertation. Research has been a period of intense learning for me, not only in the technical domain, but also on a personal level. I would like to reflect on the people who have supported and helped me so much throughout this period.

I would first like to thank and express my sincere gratitude to my supervisor Dr. Shashidhar G. Koolagudi for the constant guidance, support and encouragement at all stages of my research work. I am obliged for being introduced me to the field of machine listening. I am especially thankful to my supervisor for giving me an opportunity to pursue this research under his guidance and also for patiently checking all my manuscript and thesis. I thank him for the excellent research environment he has created for all of us to learn.

I would like to express my heartfelt thanks to my Research Progress Assessment Committee (RPAC) members Dr. B. R. Chandavarkar and Prof. T. Laxminidhi for their valuable suggestions and constant encouragement that consistently helped me in improving the research work. I would also like to thank the Head of the department and other faculty members for their help and support in carrying out research work. I would like to mention the concern and encouragement shown to me by Dr. Manu Basavarju and Dr. Basavaraj Talawar. I sincerely thank all teaching, technical and administrative staff of the Department of Computer Science and Engineering, NITK, for their help during my research work.

I am thankful to Prof. J. V. Gorabal, Prof. Sudheer Shetty, Prof. Janardhana D. R., Madesha M, Sunil B. N. and Jinto Thomas from Sahyadri College of Engineering and Management Mangaluru, for their motivation, help, and moral support given to me to pursue research at NITK, Surathkal.

I will forever remember the wonderful time I have had with my friends Prakash Pawar, Bheemappa Halavar, Khyamling Parane, Y. V. Srinivasa Murthy, Nagaratna B. Chittaragi, Pravin B. Ramteke, Girish G. N., Fathima, Pradyoth Hegde, Vishal Rathod, Anil, SivaKrishna P. M., Pramod, Kallinath, Rashmi, Saraswathi, Umapriya, Manjunatha, Uday Patil, Ningappa, Govind Giraddi, Sai Krishna, Dr. Arun Kumar, Jashma suresh, Amith and Prashanth. I am thankful to my students Jahnavi, Srujana, Akash Kademani, Chandra Church Chatterjee and many more for the new technologies we learned together. I thank all the past and present lab-mates for the friendly and conductive atmosphere in lab.

Last but not the least, a special thanks to my family. Words cannot express my sincere gratitude towards my father Shreekanth Mulimani, mother Sarojani Mulimani, brothers Mahesh and Rajesh Mulimani, Sister-in-law Meenakshi Mulimani, Nephew Manoj Mulimani, Niece Madhumati Mulimani, Panduranga Olekar, Nagartna Olekar, Shreedevi Sandimani, Rukmini Sandimani, Basavaraj Sandimani and my grand mother Laxmibai Mulimani. I am highly obligated to my family members for all sacrifices they made for my better future. Utmost thanks to the love of my life, Rashmitha. Her keen observations, clear thinking and deep rooted values in my life, have helped me overcome many a hurdles with ease. She is my strength, my mentor, my confidante and wind under my wings. Thank you dear for your love and support.

Finally I thank the almighty, for giving me the chance, the courage and the strength to endure and pursue till the end. You are the one, I always looked up to, for all my joys and trails. My trust in you will always stay firm. Thank you, God.

Place: NITK, Surathkal Manjunath Mulimani
Date:

Abstract

Currently, smart devices like smartphones, laptops, tablets, etc., need human intervention in the effective delivery of the services. They are capable of recognizing stuff like speech, music, images, characters and so on. To make smart systems behave as intelligent ones, we need to build a capacity in them, to understand and respond to the surrounding situation accordingly, without human intervention. Enabling the devices to sense the environment in which they are present through analysis of sound is the main objective of the Acoustic Scene Classification. The initial step in analyzing the surroundings is recognition of acoustic events present in day-to-day environment. Such acoustic events are broadly categorized into two types: monophonic and polyphonic. Monophonic acoustic events correspond to the non-overlapped events; in other words, at most one acoustic event is active in a given time. Polyphonic acoustic events correspond to the overlapped events; in other words, multiple acoustic events occur at the same time instance. In this work, we aim to develop the systems for automatic recognition of monophonic and polyphonic acoustic events along with corresponding acoustic scene. Applications of this research work include context-aware mobile devices, robots, intelligent monitoring systems, assistive technologies for hearing-aids and so on.

Some of the important issues in this research area are, identifying acoustic event specific features for acoustic event characterization and recognition, optimization of the existing algorithms, developing robust mechanisms for acoustic event recognition in noisy environments, making the-state-of-the-art methods working on big data, developing a joint model that recognizes both acoustic events followed by corresponding scenes etc. Some of the existing approaches towards solutions have major limitations of using known traditional speech features, that are sensitive to noise, use of features from two-dimensional Time-Frequency Representations (TFRs) for recognizing the acoustic events, that demand high computational time;

use of deep learning models, that require substantially huge amount of training data.

Many novel approaches have been presented in this thesis for recognition of monophonic acoustic events, polyphonic acoustic events and scenes. Two main challenges associated with the real-time Acoustic Event Classification (AEC) are addressed in this thesis. The first one is the effective recognition of acoustic events in noisy environments, and the second one is the use of MapReduce programming model on Hadoop distributed environment to reduce computational complexity. In this thesis, the features are extracted from the spectrograms, which are robust compared to the traditional speech features. Further, an improved Convolutional Recurrent Neural Network (CRNN) and a Deep Neural Network-Driven feature learning models are proposed for Polyphonic Acoustic Event Detection (AED) in real-life recordings. Finally, binaural features are explored to train Kervolutional Recurrent Neural Network (KRNN), which recognizes both acoustic events and a respective scene of an audio signal. Detailed experimental evaluation is carried out to compare the performance of each of the proposed approaches against baseline and state-of-the-art systems.

Keywords : Monophonic Acoustic Event Classification (AEC), polyphonic Acoustic Event Detection (AED), Acoustic Scene Classification (ASC), Time-Frequency Representations (TFRs), MapReduce programming model, Convolutional Recurrent Neural Network (CRNN), Kervolutional Recurrent Neural Network (KRNN).

Contents

List of Figures v

List of Tables ix

Abbreviations and Nomenclature xiii

1 Introduction 1

- 1.1 Monophonic Acoustic Event Classification 2
- 1.2 Polyphonic Acoustic Event Detection 3
- 1.3 Acoustic Scene Classification 6
- 1.4 Motivation 7
- 1.5 Applications 8
- 1.6 Challenges 9
  - 1.6.1 Intra-class variability 9
  - 1.6.2 Overlapping acoustic events 9
  - 1.6.3 Environmental noise and recording conditions 9
  - 1.6.4 Lack of structure 10
  - 1.6.5 Massive audio data 10
- 1.7 Highlights of the Present Research Work 10
- 1.8 Brief Overview of Thesis Contributions 11
  - 1.8.1 Spectrogram features 11
  - 1.8.2 Spectrogram image features 11
  - 1.8.3 MapReduce-based features 12
  - 1.8.4 A DNN-driven feature learning method 12
  - 1.8.5 Binaural features 12

1.8.6 Kervolutional Bidirectional Recurrent Neural Network . . . 12
1.9 Outline and Structure of the Thesis 13

2 Literature Review 15
2.1 Datasets: A Review 15
2.2 Monophonic Acoustic Event Classification: A Review 16
2.2.1 Features 23
2.2.2 Feature Representations 26
2.2.3 Classifiers 27
2.3 Polyphonic Acoustic Event Detection: A Review 30
2.3.1 Features 31
2.3.2 Classifiers 33
2.4 Acoustic Scene Classification: A Review 34
2.5 Research Gaps 35
2.6 Problem Statement 36
2.7 Common Resources used in this Work 36
2.7.1 Common Resources used for Monophonic Acoustic Event Classification 37
2.7.2 Common Resources used for Polyphonic Acoustic Event Detection 39
2.7.3 Common Resources used for the Joint Model 39
2.8 Summary 40

3 Monophonic Acoustic Event Classification 43
3.1 Monophonic Acoustic Event Classification using Spectrogram Features 43
3.1.1 Spectrogram Feature Extraction using Singular Value Decomposition 44
3.1.2 Evaluation 58
3.1.3 Contributions and Limitations 59

3.2 Robust Monophonic Acoustic Event Classification using Spectrogram Image Features 60
    3.2.1 Robust Acoustic Event Classification using Bag-of-Visual-Words 61
    3.2.2 Robust AEC using Fusion Fisher Vector Features 63
    3.2.3 Robust Acoustic Event Classification using Fusion-based Bag-of-Features 67
    3.2.4 Evaluation 68
    3.2.5 Contributions and Limitations 79
3.3 Extraction of MapReduce-based features from spectrograms for Acoustic Event Classification 80
    3.3.1 Background of MapReduce on Hadoop 81
    3.3.2 Extraction of MapReduce-based features (MRFs) from spectrogram 82
    3.3.3 Evaluation 88
    3.3.4 Contributions and Limitations 95
3.4 Summary 96

4 Polyphonic Acoustic Event Detection 97
4.1 Polyphonic Sound Event Detection using mel IFgram Features and Convolutional Recurrent Neural Network 98
    4.1.1 Extraction of mel IFgram features 98
    4.1.2 Architecture of Convolutional Recurrent Neural Network . . 99
    4.1.3 Evaluation 102
    4.1.4 Contributions and Limitations 105
4.2 A Deep Neural Network-driven feature learning method for polyphonic Acoustic Event Detection from real-life recordings 106
    4.2.1 Deep Neural Network based framework for polyphonic Acoustic Event Detection 106
    4.2.2 Evaluation 109
    4.2.3 Contributions and Limitations 110

4.3 Summary 111

5 Acoustic Scene Classification 113

5.1 Design of Convolutional Bidirectional Recurrent Neural Network for joint polyphonic Acoustic Event Detection and Acoustic Scene Classification 114

5.1.1 Evaluation 114

5.1.2 Contributions and Limitations 116

5.2 Design of Kervolutional Bidirectional Recurrent Neural Network for joint polyphonic Acoustic Event Detection and Acoustic Scene Classification 116

5.2.1 Architecture of proposed Kervolutional Recurrent Neural Network 117

5.2.2 Evaluation 118

5.2.3 Contributions and Limitations 119

5.3 Summary 119

6 Summary and Conclusions 121

6.1 Summary of the Present Work 121

6.2 Conclusions 122

6.3 Scope for Future Works 124

Bibliography 127

List of Publications 145

List of Figures

1.1 Typical monophonic Acoustic Event Classification system. Detailed descriptions are given in the text. 3

1.2 Typical polyphonic Acoustic Event Detection system. Detailed descriptions are given in the text. 4

1.3 Acoustic Events in real life scenario. Events car sound and horn are overlapped in frame t. 4

3.1 Acoustic event spectrograms. (a) acoustic event *chair moving*; (b) logarithmic spectrogram (SL) of *chair moving*; (c) resized (50×50) spectrogram of *chair moving*; (d) squared values of resized spectrogram of *chair moving*. 46

3.2 Zero crossings in left singular vectors U of *chair moving* after Singular Value Decomposition. 50

3.3 Singular vectors of chair moving obtained after Singular Value Decomposition. (a) singular vector u_1; (b) singular vector u_{N-1} (last but one). 50

3.4 Acoustic event *chair moving.* (a) singular vector u_2 (b) high zero crossings observed in u_2 ; (c) visible spikes seen in u_2; (d) singular vector u_2 without spikes. 51

3.5 Parts of singular vector u_2 of *chair moving.* (a) values selected by threshold $\pm E_t$; (b) values of u_2 not selected by threshold $\pm E_t$. . . . 55

3.6 Types of spikes. (a) Individual spikes observed in singular vector u_2; (b) Continuous multiple spikes observed in singular vector u_2. . 56

3.7 Properties of spikes present in u_2. (a) high positive value followed by a low negative extension; (b) high negative value followed by low positive extension; (c) low negative value followed by high positive extension; (d) low positive value followed by high negative extension. 56

3.8 Spectrogram feature extraction using singular vector u_2 (a1) logarithmic spectrogram of *chair moving*; (b1) spectrogram features of *chair moving*; (a2) logarithmic spectrogram of *key jingling*; (b2) spectrogram features of *key jingling*. 58

3.9 Acoustic event *Chair moving* with *speech babble* noise at 0dB SNR. (a) logarithmic spectrogram of *chair moving* at 0dB SNR; (b) spectrogram features of *chair moving* at 0dB SNR. 60

3.10 Overview of the proposed Bag-of-Visual-Words features. (a) Acoustic events; (b) grayscale spectrogram images of acoustic events; (c) transposed grayscale spectrogram images; (d) visual codebook; (e) and (f) vector quantization; (g) BoVW (histogram) representation . 61

3.11 Overview of the proposed Fusion Fisher Vector feature extraction. . 63

3.12 HSV pseudo-color mapped spectrograms of an acoustic event at clean and 0dB SNR. (*a*) acoustic event chair moving; (*b*) pseudo-color spectrogram at clean condition; (*c*) pseudo-color spectrogram at 0dB SNR; (*d*) HSV colormap. 64

3.13 Extraction of Fusion-based Bag-of-Features. (*a*) early fusion at feature level; (*b*) late fusion at representation level. 68

3.14 Average recognition accuracy of the proposed Bag-of-Visual-Words and other methods using linear, intersection and Chi-square kernels of SVM. 69

3.15 Acoustic event cup jingle. (*a*) grayscale spectrogram at clean condition; (*b*) grayscale spectrogram at 0dB SNR. 71

3.16 Recognition accuracy of proposed Bag-of-Visual-Words for different number of visual words. 72

3.17 Intensity distribution of acoustic events chair moving and laugh at clean and 0dB SNR. ($a1$) & ($b1$) spectral energy distribution of red monochrome images; ($a2$) & ($b2$) spectral energy distribution of green monochrome images; ($a3$) & ($b3$) spectral energy distribution of blue monochrome images; ($a4$) & ($b4$) spectral energy distribution of grayscale spectrogram images. 74

3.18 Acoustic event recognition accuracy of Red, Green, Blue and Fusion Fisher Vectors at clean and 0dB SNR. 75

3.19 Average recognition accuracy of proposed early and late Fusion-based Bag-of-Features at clean and different noisy conditions. . . . 76

3.20 Recognition accuracy of Fusion Fisher Vector based features versus number of Gaussians. 77

3.21 General structure of MapReduce algorithm. 82

3.22 Overview of MapReduce programming model on Hadoop framework. 82

3.23 Algorithm for identification of keypoints in large-scale logarithmic spectrograms. 83

3.24 Block diagram for identification of keypoints in large-scale logarithmic spectrograms. 84

3.25 Local MapReduce algorithm for identification of keypoints in a given spectrogram. 84

3.26 Keypoint extraction using MapReduce job. (a) logarithmic spectrogram of a gunshot, which is input to the MapReduce job; (b) identified keypoints as output from MapReduce job. 87

3.27 Comparison of runtime of MapReduce job on Hadoop and local system with sequential program for keypoint detection. (a) training phase and (b) testing phase using Mivia dataset. 90

3.28 Execution time of a MapReduce job with respect to the number of data nodes: (a) line plot shows the runtime concerning number of data nodes; (b) speedup of overall system concerning two data nodes. 91

3.29 Comparison of runtime of map and reduce phases for keypoint detection. 92
3.30 Execution time of reduce phase vis-a-vis number of reduce tasks. . . 93

4.1 Overview of proposed framework of Convolutional Recurrent Neural Network for polyphonic Acoustic Event Detection. 99
4.2 Architecture of the proposed DNN-driven feature learning framework for polyphonic Acoustic Event Detection. 106

List of Tables

2.1 Monophonic acoustic event datasets 17

2.2 Polyphonic acoustic event datasets 19

2.3 Acoustic scene datasets 21

2.4 A summery of important classification models used for monophonic Acoustic Event Classification 28

2.5 A summery of important features and Deep Neural Network models used for polyphonic Acoustic Event Detection 32

2.6 Acoustic events present in Home and Residential area scenes of 'TUT-SED 2016' development dataset. 40

2.7 Acoustic events present in ten different acoustic scenes of 'joint sound event and scene' dataset. 41

3.1 Sub-sequences of a sequence of values with exponent -10, chosen from singular vector u_2; Event: *chair moving* 53

3.2 Sub-sequences of a sequence of values with exponent -09, chosen from singular vector u_2; Event: *chair moving* 54

3.3 Sub-sequence of a sequence of values with exponent -04, chosen from singular vector u_2; Event: *chair moving* 54

3.4 Performance comparison of monophonic acoustic event recognition (in %) of the proposed Spectrogram Features with baseline system using SVM classifier. 59

3.5 Performance comparison of monophonic event recognition (in %) of the proposed BoVWs with other methods using Chi-square SVM at clean and different SNR. 70

3.6 Performance comparison of monophonic event recognition (in %) of the proposed FFV features with other methods using linear SVM at clean and different SNRs. 72
3.7 Intensity distribution distances between acoustic events at clean and 0dB noise. 75
3.8 Performance comparison of monophonic event recognition (in %) of the proposed FBoF representations using SVM with other methods at clean and varied SNRs. 78
3.9 Performance comparison of monophonic event recognition (in %) of the contribution of BoVWs with MFCCs and GTCCs using SVM at clean and different SNR on UPC-TALP dataset. 78
3.10 Performance comparison of monophonic acoustic event recognition (in %) of the proposed FFV features and FBoFs with state-of-the-art method at clean condition on UPC-TALP dataset. 79
3.11 Performance comparison of confusion matrices (in %) for monophonic acoustic event recognition on Mivia dataset using two popular and the proposed MRF-SVM approaches. 93
3.12 Performance comparison of detailed results of monophonic acoustic event recognition (in %) of two popular and the proposed MRF-SVM approaches at different SNRs. 94

4.1 Predefined set of hyperparameters for a CBRNN architecture. . . . 104
4.2 Performance comparison of polyphonic acoustic event detection using $F1$ score and error rate of the proposed mel IFgram + CBRNN method with other popular methods on the TUT-SED 2016 dataset. 104
4.3 Performance comparison of polyphonic acoustic event detection using the proposed DNN framework with other methods on the TUT-SED 2016 dataset. 109

5.1 Performance comparison of joint polyphonic AED and ASC using a combination of monaural and binaural features + CBRNN with state-of-the-art CRNN method. 115

5.2 Performance comparison of joint polyphonic AED and ASC using binaural mel IFgram and log mel band energies with KBRNN and CBRNN approach. 118

Abbreviations and Nomenclature

Abbreviations

AEC	Acoustic Event Classification
AED	Acoustic Event Detection
ASC	Acoustic Scene Classification
BoAW	Bag-of-Audio-Words
BoVW	Bag-of-Visual-Words
CNN	Convolutional Neural Network
CRNN	Convolutional Recurrent Neural Network
DNN	Deep Neural Network
ER	Error Rate
FBoFs	Fusion-based Bag-of-Features
FFV	Fusion Fisher Vector
GMM	Gaussian Mixture Models
GRU	Gated Recurrent Unit
GTCCs	Gammatone Cepstral Coefficients
HMM	Hidden Markov Models
KRNN	Kervolutional Recurrent Neural Network

MFCCs	Mel-Frequency Cepstral Coefficients
MRFs	MapReduce-based Features
NMF	Non-Negative Matrix Factorization
PCA	Principal Component Analysis
RNN	Recurrent Neural Network
SF	Spectrogram Features
SIF	Spectrogram Image Features
SVD	Singular Value Decmposition
SVM	Support Vector Machine
TFR	Time-Frequency Representations

Nomenclature

$\bigoplus$	Convolutional operation
$\bigotimes$	Kervolutional operation
$\hat{y}_t$	Predicted label vector of a frame t
F	Frequency
T	Time
u_N	N^{th} singular vector
X	Feature matrix
Y	Target output matrix
y_t	Actual label vector of a frame t

CHAPTER 1

Introduction

Billions of people, throughout the world are using personal digital assistants (PDAs). PDAs need human intervention in the effective delivery of services. Presently, some of the smart devices are able to receive information from the Global Positioning System (GPS) through satellites and then calculate the geographical location of a device/a person automatically. To make smart devices work on many other intelligent applications similar to GPS, we need to build required capacity in them, to understand the surroundings and respond to the situation accordingly without human intervention. The initial step in understanding the surroundings is capturing and processing general sound patterns prevailing in the surrounding day-to-day environment. Enabling the devices to sense the environment in which they are present through analysis of sound pattern is the main objective of Acoustic Scene Classification. Such property of intelligent devices is known as context awareness (Schilit et al., 1994).

Humans have remarkable perception ability to identify the surrounding environment based on sounds present. For instance, the sound generated from the drawer, cupboard, dishes, cutlery, water tap running, tooth brushing belong to the environment 'home'. Each sound present in an environment is known as an acoustic event. The environment in which acoustic events are present is called as an acoustic scene (Barchiesi et al., 2015). Acoustic events within an acoustic scene are broadly classified into two types. The first one is monophonic acoustic event and the other is polyphonic acoustic event. Monophonic acoustic events correspond to the non-overlapped events; in other words, at most one acoustic event is observed in a given time instance. Polyphonic acoustic events correspond to the overlapped events; in other words, multiple acoustic events occur at a given time in a mixed way. These acoustic events may be generated from human activity (for example knocking door), natural activity (for example rain sound) or vehicle noise (for example car horn).

Auditory perception capability of human beings is supported by audio-visual cues of acoustic events while understanding the scenes. Vast experience, diversified training examples, various contexts would help human beings effective and efficient classification of acoustic scenes. Acoustic scenes like a home, a park and a busy street sound are characterized by distinct acoustic characteristics, which are easily identified by human beings. On the other hand, computers still are not able to recognize the acoustic events/scenes with reliable accuracy. In this thesis, we propose and develop methods that can be used for automatic Acoustic Event Classification (AEC), Acoustic Event Detection (AED) and Acoustic Scene Classification (ASC).

1.1 Monophonic Acoustic Event Classification

ASC is a process of assigning a semantic label to an audio file that represents specific acoustic scene in the nature. AEC is a subproblem of ASC, which assigns a semantic label to an audio file that represents the particular sound event in a specific acoustic scene. The main aim of monophonic AEC is to recognize non-overlapped acoustic events in the surrounding environment (scene). An overview of monophonic AEC system is given in Figure 1.1. Digital signal processing, feature extraction and classification are the three key steps in any AEC system. 'Signal processing' phase prepares the raw acoustic signal for feature extraction. Due to computation limitations, an acoustic signal is divided into smaller frames of length, typically ranging from 10 to 30 ms before processing. Further, a window function is used to smoothen it to the required level. Unlike the Automatic Speech Recognition (ASR) system which uses the sampling frequency of 8000 Hz or lower, the AEC systems use typically a signal with high sampling frequency. Frames are

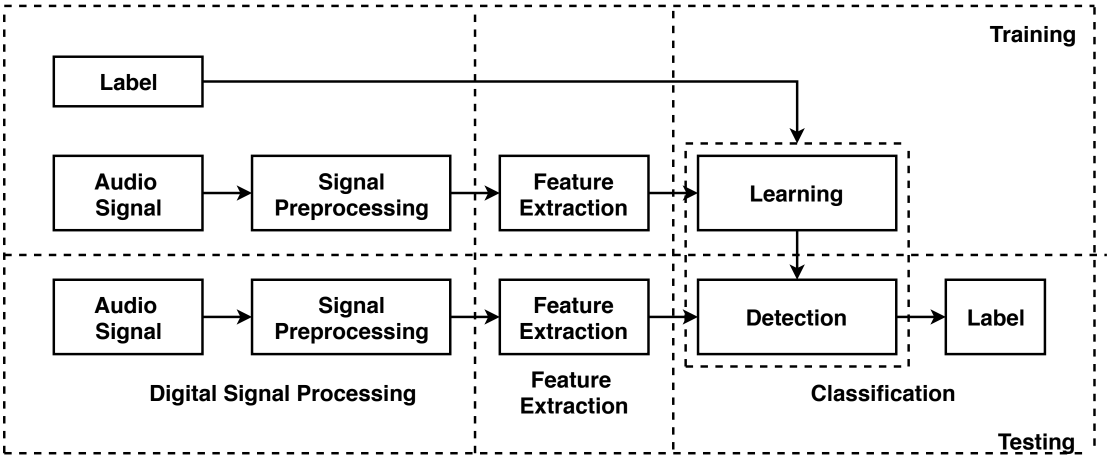


Figure 1.1: Typical monophonic Acoustic Event Classification system. Detailed descriptions are given in the text.

normally processed in overlapped manner, to avoid the loss of information around the edges of the window.

During 'feature extraction', we compress the acoustic event signal and characterize to include only the significant information. A good feature should be able to discriminate acoustic events of different classes being insensitive to surrounding noise. Generally, features are extracted from an acoustic signal frame-wise, popularly known as sequential short-time windowed frames. Each frame is represented as a feature vector. These features are also known as frame-based features. Frame-based features are commonly extracted either from the time-domain (temporal features) or frequency-domain (spectral features) of the signal. These features represent important information contained in the signal.

'Classifier' is to classify the extracted features to generate a label, to an input audio signal. The performance of the classifier is measured using recognition accuracy, which is evaluated as the percentage of correctly classified acoustic event signals to the total number of acoustic event signals present (Dennis et al., 2011; Sharan and Moir, 2015; Foggia et al., 2015).

1.2 Polyphonic Acoustic Event Detection

Polyphonic Acoustic Event Detection (AED) systems deal with the task of identifying overlapped multiple acoustic events in a continuous audio signal, in contrast to the AEC system that assigns an audio signal to one of the acoustic event classes. Polyphonic AED is more challenging and complex than monophonic AED. In every environment, an audio signal may contain both monophonic and polyphonic acoustic events in a sequence. In such situations, the detection of multiple monophonic and polyphonic acoustic events is more appropriate than typical classification.

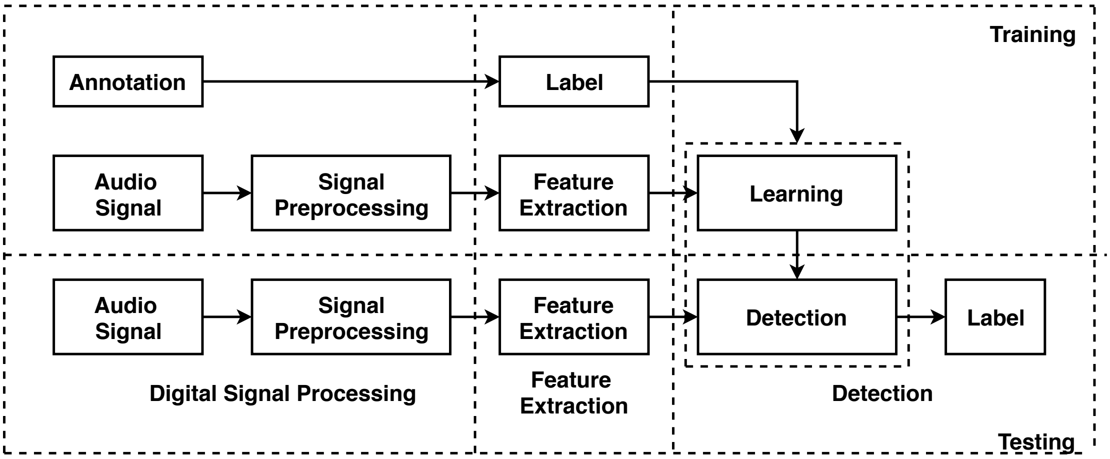


Figure 1.2: Typical polyphonic Acoustic Event Detection system. Detailed descriptions are given in the text.

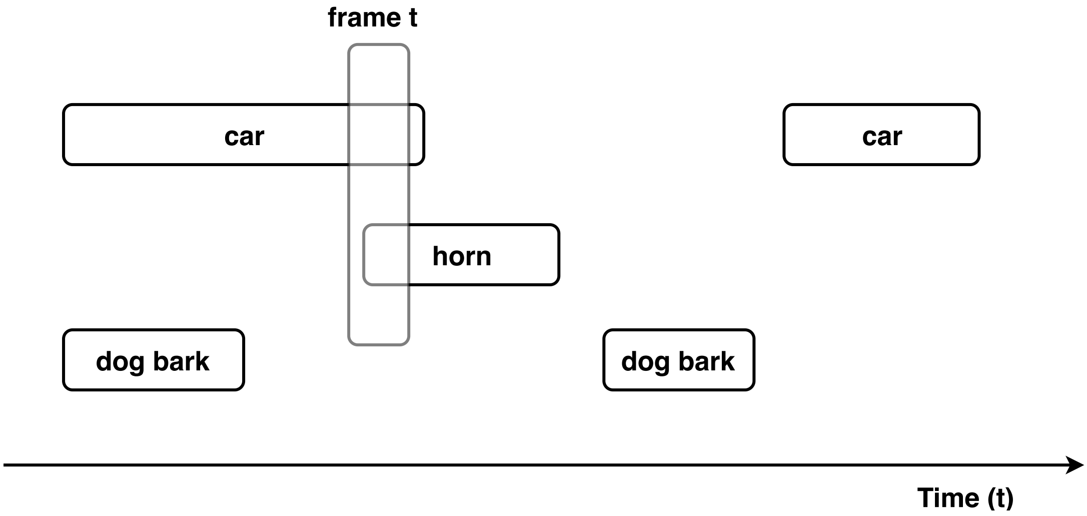


Figure 1.3: Acoustic Events in real life scenario. Events car sound and horn are overlapped in frame t.

Polyphonic AED systems (shown in Figure 1.2) also work detects frame-by-frame. Usually dataset is annotated to contain onsets, offsets and class labels associated with each event in an audio signal. A snippet of an annotation is shown in Figure 1.3, which shows the occurrences of acoustic events car, horn and dog barking along the time domain. One can observe that acoustic events car and horn are overlapped in frame 't'. In fact polyphonic AED systems are expected to detect both isolated and overlapped acoustic events in an audio signal. The training stage of the system (shown in Figure 1.2) maps the labels taken from annotation to particular acoustic event instances frame-by-frame. In this context, labels are also known as acoustic event activity indicators. Unlike one-dimensional label vector in the AEC system, here label vector is of dimension equal to the number of acoustic events in the dataset. If the frame contains an event/events, then value/values of label vector are set to 1 otherwise set to 0.

In the testing phase, the trained system decides a label for each frame of the test audio signal. In the case of polyphonic AED, performance of the system is evaluated by comparing the output with the reference annotation. There are no universally accepted metrics for polyphonic AED. However, commonly used metrics for performance evaluation of polyphonic AED system are broadly classified into two types: segment-based metrics and event-based metrics (Mesaros et al., 2016a). Segment-based metrics compare the output of a system and corresponding reference in a fixed length interval, known as a segment. Event-based metrics compare the output of a system and reference event by event. Two segment-based metrics popularly used for evaluation of the performance of polyphonic AED system are $F1$-score and Error Rate (ER) (Mesaros et al., 2016a). The following intermediate statistics are computed for evaluation of $F1$-score and Error Rate (ER) per segment.

- True Positive ($TP(p)$): The total number of acoustic events detected correctly by the system from the test (reference) segment p.

- False Positive ($FP(p)$): The total number of wrongly detected acoustic events by the system from the test segment p.

- False Negative ($FN(p)$): The total number of acoustic events present in the test segment p but not detected by the system.

$F1$ score is considered as the first metric and computed using intermediate statics as given in equation (1.1),

$$F1 = \frac{2 \times \sum_{p=1}^{P} TP(p)}{2 \times \sum_{p=1}^{P} TP(p) + \sum_{p=1}^{P} FP(p) + \sum_{p=1}^{P} FN(p)} \tag{1.1}$$

Where p is a specific segment and P is the total number of segments. Another set of intermediate statics namely, Insertion (I), Deletion (D) and Substitution (S) are computed for each segment p.

- $I(p)$: The number of false positives, which are not considered as substitutions.
- $D(p)$: The number of false negatives, which are not considered as substitutions.
- $S(p)$: The number of events in the test (reference) segment for which system detects wrong events.

The Error Rate (ER) is computed as the second metric using intermediate statistics I, D and S as given in equation (1.2),

$$ER = \frac{\sum_{p=1}^{P} S(p) + \sum_{p=1}^{P} D(p) + \sum_{p=1}^{P} I(p)}{\sum_{p=1}^{P} M(p)} \tag{1.2}$$

Where $M(p)$ is the number of acoustic events in the test segment p.

1.3 Acoustic Scene Classification

An ASC system assigns input audio signal to one of the acoustic scene classes in the dataset and its general structure is the same as the monophonic AEC system. However, in this thesis, acoustic scenes are characterized based on acoustic events present in them. As we had mentioned earlier, each acoustic scene has its own set of acoustic events. In this thesis, polyphonic AED system is used to detect both acoustic events (either monophonic or polyphonic) and a corresponding scene present in an input audio signal concurrently. This type of model is also known as a joint polyphonic acoustic event detection and scene recognition system (joint model).

The joint model predicts acoustic events and respective scenes frame-by-frame from the input audio. Acoustic scene label of an audio file represents an acoustic scene present in the majority of the frames of an audio recording (majority voting). The recognition performance of polyphonic AED and ASC of the joint model is evaluated using $F1$-score, ER and majority voting accuracy (Bear et al., 2019)

1.4 Motivation

Research on AEC is still in its infancy compared to the other audio processing tasks such as speech processing. Speech is different from acoustic events when one considers the phonetic structure. The traditional frame-based speech features such as Mel- frequency cepstral coefficients (MFCCs) are specifically obtained for speech/speaker recognition tasks which may not be suitable for AEC because of varying acoustic and phonetic properties. Unlike speech, acoustic events are short in duration and have more distinct Time-Frequency Representations (TFRs) (Dennis et al., 2011). This motivates to think of novel techniques for extraction of acoustic event specific features.

Recently in the literature, features extracted from spectrogram are said to be robust for AEC in noisy conditions. This motivates us to extract more reliable features from the spectrograms for AEC using different methods. Feature extraction from two-dimensional spectrograms of a large noisy audio event dataset demands high computational time. In this context, parallelizing the feature extraction task using MapReduce programming model on Hadoop setup improves the efficiency of the overall system.

DNN (Deep Neural Networks) models are widely used for polyphonic AED in the literature. However, these DNN models require a larger dataset for training. If the dataset is not sufficiently large, then these models encounter a problem such as overfitting. This motivates us to develop DNN-driven feature learning method for polyphonic AED, that should consider smaller real-time datasets also.

AEC/AED is the subproblem of ASC. Majority of the approaches reported in the literature either concentrated on AEC/AED or ASC and developed them as two independent systems. This motivates us to build a single system which identifies/classifies both acoustic events and acoustic scenes. This system would be more reliable and time-efficient than individual AEC/AED and ASC systems.

1.5 Applications

Some of the important applications of AEC/AED and ASC are: assistive technologies that would help the hearing impaired in their daily lives (Xiang et al., 2010); intelligent healthcare systems such as identification of cough sounds of patients (Peng et al., 2009)(Goetze et al., 2012); context-aware devices such as smartphones that continuously sense their surroundings and take necessary actions like switching to silent mode every time a person enters a meeting room, robotic wheelchairs that adjust their functioning based on the recognition of indoor or outdoor environments; driver-less cars that adjust their speed based on recognition of acoustic events in a city-center or a quiet street; recognition of the background environment in the crime spot for forensics; and so on.

Recently, there are AEC systems reports of using for the purpose of audio-based surveillance. Audio-based surveillance is an important aspect of safety, security and monitoring applications such as human activities monitoring system (Harma et al., 2005), Wildlife monitoring system (Somervuo et al., 2006), smart homecare system(Chen et al., 2013), hazardous acoustic event recognition in the elevator (Radhakrishnan et al., 2005), on the road recognition of acoustic environment (Foggia et al., 2016) and so on. Early surveillance systems relied solely on cameras (video sensors). These video cameras have poor performance in darkness and are sensitive to weather conditions, shadow, reflection, illumination etc(Crocco et al., 2016)(Räty, 2010). The visual information from the camera has its own limitations. For instance, the dangerous event such as gunshot has distinct acoustic characteristics, which is not easily identified and captured by either video or an image. Nowadays, IP cameras are available with audio sensors (microphones), which allows analyzing both audio and video streams for improved security (Cristani et al., 2007). Unlike cameras, microphones can record sounds even in darkness and cover larger distances with a cheaper development cost.

1.6 Challenges

Research progress on ASC remained stagnant until recent years. This is due to several challenges faced in a real-time scenarios as it is not possible to identify all acoustic events with better accuracy. Some of the important challenges are explained below in brief.

1.6.1 Intra-class variability

Acoustic event classes being broadly used for AEC/AED systems are defined such as key jingling, a phone ringing, vehicle horn and so on, causing high rate of intra-class variability. For instance, an acoustic event class 'horn' represents all types of vehicle horns, whose acoustic characteristics significantly vary among themselves. An AEC system should be able to detect vehicle horns with higher specificity such as bus and lorry.

1.6.2 Overlapping acoustic events

Earlier AEC/AED systems mainly focused on the recognition of acoustic events that were non overlapped. However, in the real-time scenario, acoustic events often occur at the same time instances mostly in an overlapped way. For instance, a recording from a home scene may include water-tap running, people speaking, television sound and foot-steps; all occurring at the same time. To improve their efficiency, the AED systems should be able to discriminate the acoustic characteristics of each individual event available as the combined acoustic mixture.

1.6.3 Environmental noise and recording conditions

The scope of AEC is defined as the set of pre-identified acoustic event classes. The acoustic events that are not in the scope of AEC, are obviously considered as background noises. For instance, wind sound is a common noise present in the real-time recordings and it reduces the signal-to-noise ratio (SNR) significantly.

In addition, variations present in the recording conditions such as quality of the recording devices and distance of the recording device from the sound sources cause additional challenges.

1.6.4 Lack of structure

Other acoustic signals such as speech and music have a formal structure, which is used to extract informative sound representations from a signal. For instance, speech can be decomposed into phonemes and the properties of each phoneme can be investigated. It is convenient to map phoneme to its language representation and can be used in speech recognition. Similarly, music can be divided into notes and be used as units of processing. On the other hand, it is highly difficult to represent (divide) the acoustic events in a standard structure and characterize them. This is due to the simple reasons that; Acoustic events simultaneously contribute to the different acoustic scenes and are generally generated from various sound sources. This makes AEC task more challenging than other audio processing tasks.

1.6.5 Massive audio data

Thousands of audio sensors (recording devices) deployed in the different acoustic scenes for environment monitoring, generate massive audio data. Processing this big data high computational resources. State-of-the-art methods on AEC task recognize acoustic events by extracting features from the smaller dataset and may not be suitable for real world and real time applications.

1.7 Highlights of the Present Research Work

- A comprehensive analysis of the literature on the methods of monophonic AEC, polyphonic AED and ASC including datasets.

- Proposing spectrogram and spectrogram image based features for monophonic AEC.

- Proposing two deep learning-based models; one is improved Convolutional Recurrent Neural Network (CRNN) and the another is a DNN-driven feature learning approach for polyphonic AED.
- Developing a joint model that recognizes both acoustic events and corresponding scene.
- Proposing a combination of binaural features and Kervolutional Bidirectional Recurrent Neural Network (KBRNN) for joint polyphonic AED and ASC.

1.8 Brief Overview of Thesis Contributions

The major contributions of this thesis include exploring the acoustic event specific features from the spectrogram for monophonic AEC and polyphonic AED. Two novel monophonic AEC methods and one novel polyphonic AED method are proposed. The brief details are given in the following sub-sections.

1.8.1 Spectrogram features

Normally, acoustic events have distinct Time-Frequency (TF) representations. It hints that visual information from spectrogram may be promising features for AEC. High-energy spectral components of the acoustic events are extracted as features from the spectrogram using Singular Vector Decomposition (SVD) (Mulimani and Koolagudi, 2019c) and MapReduce programming paradigm (Mulimani and Koolagudi, 2019a); for monophonic AEC task. Experiments show that proposed spectrogram features outperform traditional speech features.

1.8.2 Spectrogram image features

Spectrogram images represent visual information. In this work, spectrogram image of an acoustic event is used as a fixed dimensional feature vector known as Bag-of-Visual-Words (BoVWs). BoVWs along with Fisher kernel encoding methods are named as spectrogram image features (Mulimani and Koolagudi, 2018).

Spectogram image based features are better compared to the spectrogram features (explained in 1.8.1) extracted directly from spectrograms. The combination of different feature representations is used to get robust features for AEC. Experiments show that the proposed features are robust to noise and outperform state-of-the-art methods in both clean and noisy conditions.

1.8.3 MapReduce-based features

Extraction of reliable, task specific information as features from spectrograms of big noisy audio event dataset demands high computational resources. Parallelizing the feature extraction using the MapReduce programming model on Hadoop improves the time efficiency of the overall system. A parallel method is proposed for extraction of significant information related to the event from a spectrogram using Google's MapReduce programming model (Dean and Ghemawat, 2008). These features are known as MapReduce-based features (MRFs).

1.8.4 A DNN-driven feature learning method

A series of layers including two projection layers, a CNN layer, two fully connected layers and a sigmoid layer are stacked to construct the proposed DNN-driven feature learning method for polyphonic AED. New projection layers and CNN layer learn the discriminative spectral properties of multiple overlapped acoustic events in the mixture effectively and outperform state-of-the-art methods.

1.8.5 Binaural features

Polyphonic acoustic events and a respective scene of an audio signal recognized better with features from multi-channels. Different combinations of binaural features are explored to train DNN models effectively.

1.8.6 Kervolutional Bidirectional Recurrent Neural Network

The performance of the CNN in CRNN may be further improved by generalizing the convolutional operation to non-linear operation using a polynomial kernel.

This architecture is also known as a kervolutional Neural Network (KNN). A combination of KNN and bidirectional Gated Recurrent Unit (GRU) forms KBRNN for joint polyphonic AED and ASC.

1.9 Outline and Structure of the Thesis

The thesis is spread across 6 chapters. The following paragraphs broadly explain the contents of each chapter.

- **Chapter 1 : The Introduction** covers introduction of the task of ASC. Monophonic and polyphonic acoustic events in an acoustic scene are briefly discussed. Motivation, applications, challenges during recognition of acoustic events and scenes are briefly discussed. Chapter ends with the clearly articulated research contributions and thesis outline.

- **Chapter 2 : Literature review** mainly contains the list of available datasets and their basic properties for monophonic AEC, polyphonic AED and ASC. Further, this chapter contains information about primitive state-of-the-art methods of monophonic AEC, polyphonic AED and ASC in the context of features and classifiers. Research gaps are identified, enumerated and discussed. The common datasets used in this research work are introduced. Scope of the present work derived from the literature review is presented.

- **Chapter 3 : Monophonic Acoustic Event Classification** includes the proposed spectrogram and spectrogram image based features for monophonic AEC. Detailed experiments carried out to analyze the robustness of the proposed methods in noisy conditions are presented. Results are discussed with appropriate analysis and conclusions.

- **Chapter 4 : Polyphonic Acoustic Event Detection** includes the explanation of improved CRNN and the DNN-driven feature learning approach for polyphonic AED from real-life recordings. The significance of each of the models is discussed. The performance of the proposed approaches is presented along with the necessary analysis and discussion.

- **Chapter 5 : Acoustic Scene Classification** includes binaural features for improved CRNN and KBRNN models for joint polyphonic AED and ASC. The performance of both models is presented with appropriate analysis and discussion.
- **Chapter 6 : Summary and conclusions** chapter summarizes the contributions of this thesis along with some important conclusions. This Chapter also provides possibilities of the extensions to the present work and future research directions for improving the performance of AEC/AED and ASC models.

CHAPTER 2

Literature Review

In chapter 1, an overview of monophonic Acoustic Event Classification (AEC), polyphonic Acoustic Event Detection (AED) and Acoustic Scene Classification (ASC) was provided, with a discussion on motivation, potential applications and challenges faced in recognition of acoustic events present in an acoustic scene. The aim of this chapter is to give deeper insight into the range of recent approaches developed and reported in the literature specifically for monophonic AEC, polyphonic AED and ASC concerning datasets, features and classifiers. A list of the research gap gaps is derived from the critical review of the available literature at the end of the chapter.

2.1 Datasets: A Review

A suitable acoustic event/scene dataset is a necessary for acoustic event/scene classification. The design and collection of acoustic event dataset mainly depend on the research applications. For instance, a dataset with acoustic events such as gunshot, glass breaking and person screaming are used in audio-based surveillance. Similarly, a dataset with acoustic events applause, chair moving, laugh, door knock, etc. are used to study/analyze the meeting room scene. The survey presented in this section introduces the publicly available acoustic event and scene datasets.

Popularly used monophonic acoustic event datasets are listed in the Table 2.1. Each dataset is designed and developed for particular application. However, a good AEC system should be independent of the input dataset. In this thesis, UPC-TALP dataset and Mivia audio event dataset are chosen for performance evaluation of the proposed monophonic AEC approaches. However, any dataset from Table 2.1 can be used for performance evaluation. UPC-TALP dataset is developed and released as a part of the CHIL (Computers in the Human Interaction Loop) acoustic event detection challenge (Temko et al., 2006a). Mivia audio event dataset is a real-time dataset, developed for audio-based surveillance.

Popularly used polyphonic acoustic event datasets are listed in Table 2.2. Majority of them are TUT sound event datasets, which are developed and released as parts of different editions of DCASE challenge. In this thesis, TUT Sound Events 2016 dataset is used for performance evaluation of the proposed polyphonic AED approach. TUT Sound Events 2016 dataset includes real-time acoustic events from home and residential areas.

Widely used acoustic scene datasets are listed in Table 2.3. In this thesis, the acoustic scene is recognized based on the acoustic events present in it. For this task, joint sound scene and event dataset is considered as an input to the proposed ASC system, which characterizes both acoustic events and scenes.

2.2 Monophonic Acoustic Event Classification: A Review

AEC problem may be formulated as a machine learning problem that consists of two main stages. One is the feature extraction and other is classification (shown in Figure 1.1). As we had mentioned in chapter 1, in the feature extraction stage, conventional systems extract the fixed-dimensional features from the acoustic signal mostly frame-by-frame. Such features are also known as frame-based features and further they are used for classification. The most commonly used frame-based features are Mel-Frequency Cepstral Coefficients (MFCCs). In the classification stage, the classifier learns to recognize the acoustic events using extracted features. The different types of classifiers are used for AEC.

Table 2.1: Monophonic acoustic event datasets

Sl. No.	Dataset	No. of recordings	No. of events	Purpose	References
01	Real World Computing Partnership (RWCP) sound dataset	4800	60	Collision, action and characteristics sounds used for acoustic event recognition	(Nakamura et al., 2000)
02	University of Catalunya - Center for Language and Speech Technologies and Applications (UPC-TALP) dataset	approximately 840	14	Acoustic events applause, chair moving, laugh, door knock, etc. are used to monitor the meeting room scene	(Temko and Nadeu, 2009)
03	Fondazione Bruno Kessler (FBK) dataset	approximately 800	16	Acoustic events applause, chair moving, laugh, door knock, etc. are used to monitor the meeting room scene	(Cotton and Ellis, 2011)
04	Kitchen dataset	1526	24	Acoustic events water tap running, plate sorting, tooth brushing, etc. are used to monitor the activity of human in kitchen	(Stork et al., 2012)
05	Urban sound 8k dataset	8732	10	Acoustic events car horn, children playing and so on are used for Urban sound analysis in smart cities	(Salamon et al., 2014)
06	NAR (developed using NAo Robot) dataset	852	42	Acoustic events from different scenarios are used in robotics	(Maxime et al., 2014)
07	Mivia audio event dataset	18000	3	Acoustic events glass breaking, gun shots and person screaming used for audio-based surveillance	(Foggia et al., 2015)

Table 2.1: Monophonic acoustic event datasets

08	Environmental Sound Classification (ESC) dataset	2000	50	Combination of animal sounds natural sound scapes, human sounds indoor and outdoor sounds used for AEC	(Piczak, 2015b)
09	Mivia road events dataset	400	2	Acoustic events car crashes and tire skidding are used for road surveillance	(Foggia et al., 2015)
10	IEEE Audio and Acoustic Signal Processing (AASP) challenge on Detection of Acoustic Scene and Events (DCASE)-2013 dataset	960	16	Acoustic events printer, mouse click, drawer, etc., are used to monitor the office	(Stowell et al., 2015)
11	FINCA (a smart conference room at TU Dortmund University) dataset	437	19	Acoustic events are used to monitor the conference room	(Kürby et al., 2016)

Table 2.2: Polyphonic acoustic event datasets

Sl. No.	Dataset	Length of data (in minutes)	No. of events	Purpose	References
01	Tampere University of Technology Sound Events Detection (TUT-SED) 2009 dataset	1133	61	Acoustic events from ten different acoustic scenes are used for detection of polyphonic acoustic events.	(Heittola et al., 2010)
02	Computational Hearing in Multisource Environments (CHiME)-Home dataset	408	7	This dataset is used for sound source recognition in a home environment.	(Foster et al., 2015)
03	Tampere University of Technology Sound Events Detection (TUT-SED) 2016 dataset	78	20	Real-time home and residential area acoustic events used for detection of polyphonic acoustic events as a part of Detection and Classification of Acoustic Scenes and Events (DCASE)-2016 challenge.	(Mesaros et al., 2016b)
04	Tampere University of Technology Sound Events Detection (TUT-SED) synthetic 2016 dataset	566	16	Synthetic mixture of acoustic events used for detection of polyphonic acoustic events as a part of Detection and Classification of Acoustic Scenes and Events (DCASE)-2016 challenge.	(Mesaros et al., 2016b)

Table 2.2: Polyphonic acoustic event datasets

05	Tampere University of Technology Sound Events Detection (TUT-SED) 2017 dataset	70	6	Real-time street acoustic events used for detection of polyphonic acoustic events as a part of Detection and Classification of Acoustic Scenes and Events (DCASE)-2017 challenge.	(Mesaros et al., 2017)
06	Joint sound event and scene dataset	1500	32	Synthetic mixture of acoustic events from ten acoustic scenes are used for polyphonic acoustic event detection	(Bear et al., 2019)

Table 2.3: Acoustic scene datasets

Sl. No.	Dataset	No. of recordings	No. of scenes	Acoustic scenes	References
01	Tampere University of Technology (TUT) Computational Auditory Scene Analysis (CASA)-2009 dataset	103	10	basketball, beach, bus, car, hallway, office, restaurant, shop, street, track & field	(Heittola et al., 2010)
02	Rouen university dataset	3026	19	busy street, bus, cafe, car, train station, kid game hall, market, metro-paris, metro-rouen, pool hall, quiet street, student hall, restaurant, pedestrian street, shop, train, high speed train, tubestation	(Rakotomamonjy and Gasso, 2015)
03	IEEE AASP (Audio and Acoustic Signal Processing) Challenge 2013 dataset	100	10	busy street, quiet street, Park, open-air market, bus, subway-train, restaurant, shop/supermarket, office, subway station	(Stowell et al., 2015)
04	Tampere University of Technology (TUT) acoustic scenes 2016 development dataset	1170	15	beach, bus, cafe, car, city, forest, grocery, home, library, metro, office, park, residential area, train, tram	(Mesaros et al., 2016b)
05	Tampere University of Technology (TUT) acoustic scenes 2017 development dataset	4680	15	beach, bus, cafe, car, city, forest, grocery, home, library, metro, office, park, residential area, train, tram	(Mesaros et al., 2017)

Table 2.3: Acoustic scene datasets

06	Tampere University of Technology (TUT) urban acoustic scenes 2018 development dataset	8640	10	airport, bus, metro, metro station, park, public square, shopping mall, pedestrian, traffic, tram	(Mesaros et al., 2018)
07	Tampere University of Technology (TUT) urban acoustic scenes 2018 mobile development dataset	10080	10	airport, bus, metro, metro station, park, public square, shopping mall, pedestrian, traffic, tram	(Mesaros et al., 2018)
08	Joint sound event and scene dataset	3000	10	bus, busy street, office, open air market, park, quite street, restaurant, supermarket, tube, tube station	(Bear et al., 2019)

However, the performance of a classifier is dependent on the significance of the features used for classification (Kons et al., 2013). It is to be noted here that, identification and extraction of effective features for the development of AEC system is really a challenging task.

2.2.1 Features

Majority of the feature extraction techniques reported in the literature depend on the four broad domains of acoustic signal representation, namely: temporal, spectral, cepstral and joint Time-Frequency representations. Most commonly used temporal features for AEC are the signal energy and zero-crossing rates (Chu et al., 2009; Temko and Nadeu, 2006), whereas commonly used spectral ones are the spectral flux, spectral slope, spectral centroid, spectral flatness and spectral rolloff (Perperis et al., 2011; Temko and Nadeu, 2006; Kim and Ko, 2011; Zhang and Schuller, 2012; Lojka et al., 2016; Maxime et al., 2014). Temporal and spectral features are mainly used as supplementary features to each other.

Cepstral features are also referred to as cepstral coefficients. Inverse Fourier Transform (IFT) of the log of magnitude spectrum of a signal gives cepstrum. Cepstrum represents variations of frequency components in a spectrum. Hence, it is also known as spectrum of spectrum. Probably, Linear Prediction Cepstral Coefficients (LPCCs) are the oldest cepstral domain features (Makhoul, 1975), which are later replaced by the MFCCs (Davis and Mermelstein, 1980). Human beings are able to identify even small frequency variations in lower ranges than higher once. MFCCs closely resemble human perception system and are derived from mel-filters, which equally space frequency bands as per the mel-scale (Stevens et al., 1937). The first (Δ) and second ($\Delta\Delta$) order derivatives of MFCCs are concatenated with the MFCCs to improve the performance of AEC system (Young et al., 2009).

Recently, Gammatone Cepstral Coefficients (GTCCs) are added to the family of cepstral domain features. GTCCs are derived from Gammatone filters (Slaney et al., 1993), which model the frequency selection property of human cochlea (Patterson et al., 1992). Filters are equally spaced on the Equivalent Rectangular

Bandwidth (ERB) scale. GTCCs are effectively used for ASR (Cheng et al., 2005) and AEC (Valero and Alias, 2012).

The feature or evidence level combination of temporal, spectral and cepstral features is also used to implement the AEC system (Foggia et al., 2015; Kiktova-Vozarikova et al., 2015; Zhuang et al., 2010; Foggia et al., 2016; Maxime et al., 2014). Temporal, spectral and cepstral features are frame-based ones and specifically designed, computed and evolved for speech/speaker recognition tasks. These features are designed to extract acoustic characteristics of speech, which are quite different from that of acoustics event and may not be suitable for AEC.

Acoustic events have more distinct TF features (characteristics) than speech. Such features are commonly extracted from the spectrograms of the acoustic signals. There are two different approaches widely used for this task. One is Non-Negative Matrix Factorization (NMF), which first decomposes the spectrogram (or equivalent TFRs) into the base and coefficient vectors followed by feature extraction from the decomposed vectors (Ghoraani and Krishnan, 2011). Such NMF-based features perform better than traditional frame-based features for AEC (Ludeña-Choez and Gallardo-Antolín, 2016). However, we cannot control the outcome of factorization (Heittola et al., 2011). Hence, the output of NMF is not unique each time we run on real-time data (Ghoraani and Krishnan, 2011). This is a serious issue. The other approach reported in (Dennis et al., 2013b), directly extracts the features from the spectrograms region-by-region. These features are more specific to the acoustic event. However, the extraction of features from each individual region demands high computational time and impractical on larger dataset. Biologically-inspired two-dimensional Gabor-filter functions are used to capture spectro-temporal modulations of acoustic events (Schröder et al., 2015), which are also computationally expensive. Recently, Deep Neural Networks (DNNs) are used for AEC (Kong et al., 2016).

Generally, real-time acoustic events are available overlapped with high-background noise. Traditional frame-based features are sensitive to noise and their performance degrades as SNR decreases (Dennis et al., 2011). Recently, features from the spectrogram image are proved to be effective for robust AEC in noisy conditions (Dennis et al., 2011; Sharan and Moir, 2018, 2015). A spectrogram is computed using Short-Time Fourier Transform (STFT), where acoustic event signal is partitioned into frames of a specific length and Discrete Fourier Transform (DFT) is applied to get spectra. These spectra of complex values are concatenated side-by-side to form spectrogram. However, most of the classifiers are designed to work only with real-valued input. Hence, less informative phase information is discarded (Gerhard, 2003), and only the magnitudes of the spectrogram retained to form magnitude spectrogram. Besides, the log is taken to reduce the dynamic range of values of magnitude, resulting in log magnitude spectrogram. Different variations of spectrograms are also reported in the literature (Hlawatsch and Boudreaux-Bartels, 1992; Patterson et al., 1992; Heil and Walnut, 1989). Gammatone spectrogram (also known as cochleagrams) is one of them, computed using Gammatone filter. Gammatone spectrograms represent higher intensity values of an acoustic event more clearly than a conventional STFT-based spectrogram. Unlike STFT-based spectrogram narrow bandwidths at the lower frequency regions and wider bandwidth at higher frequency regions are used for Gammatone spectrogram construction (Sharan and Moir, 2015).

Dennis et al. (2011) converted the spectrogram into a pseudo-color spectrogram image. Monochrome images of the pseudo-color spectrograms are divided into blocks. Second and third order central moments are evaluated from each block and used as Spectrogram Image Features (SIFs). SIFs were shown to be more robust and performed significantly better than MFCCs in different noisy conditions. However, use of second central moments from image blocks leads to the significant loss of information from the spectrogram image. In (Sharan and Moir, 2018), the features are selected from Gammatone spectrogram images using Sequential Backward Feature Selection (SBFS) algorithm and used for AEC. However, SBFS is a greedy algorithm, that demands high computational time and impractical on the larger dataset.

From the available literature on features, it is fairly clear that frame-based speech related features may not be suitable for AEC. Hence there is a scope to design and extract different features that can better capture the acoustic event information. The features from spectrogram are shown to be acoustic event specific and robust to noisy conditions. The state-of-the-art methods to extract features from the spectrograms demand high computational time and have their own disadvantages. There is a scope to develop a time-efficient approaches for feature extraction from spectrograms, especially during the use of massive audio datasets available noisy conditions.

2.2.2 Feature Representations

Due to variations in the length of the input acoustic event signals (audio clips), frame-based feature extraction techniques give a different number of fixed-dimensional feature vectors. If an acoustic event is represented by a sequence of feature vectors, then the number of feature vectors in different acoustic events are different based on the length of the given acoustic event. Such variable lengthed sequences are effectively modeled using Gaussian Mixture Model (GMM) based Hidden Markov Models (HMMs) (Dennis et al., 2013a). However, HMMs have two major issues. One is, HMMs require huge training data to capture distinct variations among acoustic events making them less fit for simple and compact applications of the modern digital world. Other is HMMs map the feature vector frame-by-frame from a speech signal into its intermediate semantic label such as phoneme, syllable, etc., before mapping it to a corresponding utterance. Acoustic events do not have such intermediate labels. Therefore, mapping of feature vectors to their event classes directly may cause confusions in the models.

Recently, Support Vector Machine (SVM) classifier is shown to be highly effective for AEC (Temko et al., 2006b; Sharan and Moir, 2018). However, SVM requires a fixed-dimensional sequence of feature vectors. Several feature representation (learning) methods are map the variable length sequences into fixed-length ones. A most common method is use of features obtained by evaluating the statistical parameters such as mean, standard deviation, mode and median of features from every frame of an acoustic signal. This approach transforms the frame-based unequal lengthed feature vectors into a fixed length feature vector, causing unavoidable information loss (Guo and Li, 2003). Popular advanced representation methods reported in the literature are the Bag-of-Audio-Words (BoAW) (Pancoast and Akbacak, 2012) and the Fisher kernel (Temko et al., 2006b) approaches. BoAW approach represents the frame-based features into a fixed-dimensional histogram ('bag') known as BoAW. This histogram is used as a feature vector to SVM. Recently, it is reported that BoAW approach even outperforms the popular Deep Neural Network (DNN) based classification (Schmitt et al., 2016)(Grzeszick et al., 2017). Alternatively, Fisher kernel (Jaakkola and Haussler, 1999) is also used to represent frame-based features into a fixed-dimensional feature vector, known as the Fisher vector. A generalized Fisher kernel named as score-space kernel is used for speech recognition (Smith and Gales, 2002), speaker verification (Wan and Renals, 2005) and AEC (Temko et al., 2006b). As we had mentioned, traditional frame-based features are speech specific and sensitive to noise, even Fisher vector/BoAW representations of frame-based features may not be suitable for AEC both in clean and noisy conditions.

2.2.3 Classifiers

Classifiers used for AEC are broadly classified into two types statistical models and deep learning models. A summery of some of the classification models used for monophonic AEC task are listed in the Table 2.4.

Statistical models such as Support Vector Machine (SVM) (Foggia et al., 2015; Guo and Li, 2003; Dennis et al., 2011; Phan et al., 2016a; Jayalakshmi et al., 2018), Gaussian Mixture Models (GMM), Hidden Markow Models (HMM) (Maxime et al., 2014), K-Nearest Neighbor (KNN), Random Forest (RF) (Piczak, 2015b) etc., are used for monophonic AEC in the literature. Recently, deep learning methods such as Feed-Forward Neural Networks (FFNN) (McLoughlin et al., 2015), Convolutional Neural Networks (CNN) (Piczak, 2015a)(Salamon and Bello, 2017), Recurrent Neural Networks (RNN) (Freitag et al., 2017) are successfully applied for monophonic AEC.

Table 2.4: A summery of important classification models used for monophonic Acoustic Event Classification

Sl. No.	Models	Features	Feature representations	Dataset	References
01	SVM	SIFs	Statistical representation	RWCP	(Dennis et al., 2011)
02	HMMs	MFCCs+NMF		FBK	(Cotton and Ellis, 2011)
03	GMMs, HMMs	Temporal+spectral+MFCCs	-	NAR	(Maxime et al., 2014)
04	SVM, KNN	Temporal+spectral+MFCCs	BoAW	NAR	(Maxime et al., 2014)
05	Maximum Likelihood	MFCCs+GTCCs	BoAW	FINCA, DCASE-2013	(Plinge et al., 2014)
05	SVM	MFCCs	Statistical representation	Urban sound 8k	(Salamon et al., 2014)
06	SVM	Temporal+spectral	BoAW	Mivia	(Foggia et al., 2015)
07	SVM, KNN, RF	Temporal+MFCCs	Statistical representation	ESC	(Piczak, 2015a)
08	HMMs	TF Gabor features	-	DCASE-2013	(Schröder et al., 2015)
09	SVM	Spectral+Temporal	BoAW	UPC-TALP, Kitchen, NAR	(Phan et al., 2016a)
10	SVM	MFCCs	Statistical representation	ESC, DCASE-2013	(Jayalakshmi et al., 2018)
11	FFNN	Spectrogram image	TFR	RWCP	(McLoughlin et al., 2015)
12	CNN	Mel-spectrogram	TFR	Urban sound 8k	(Salamon and Bello, 2017)
13	RNN	Spectrogram	TFR	ESC	(Freitag et al., 2017)

1 to 10 are statistical models
11 to 13 are deep learning models

Performance of the DNN based models is reported to be better and treated as state-of-the-art in the field of computer vision (Deng et al., 2019), speech processing (Cauchi et al., 2019), speech enhancement (Pandey and Wang, 2019), machine translation (Wang et al., 2018), music classification (Choi et al., 2017) and so on.

Information processing by DNNs is somewhat similar to that of human brain. A series of interconnected neurons (layers) are stacked to construct DNN. The parameters of each neuron (wigths and biases) are iteratively updated through gradient descent optimization method, which minimizes the cost (error) between actual and predicted outputs. A first layer that receives input is known as the input layer. The last layer that predicts the output of a network is known as the output layer. The intermediate layers between the first and last layers are known as hidden layers. The network architecture and training procedure of a DNN are defined by setting hyper-parameters such as number of layers, number of hidden units in layers, regularization parameters, optimization parameters and so on (Cakir et al., 2017).

Generally, Different variations of the spectrograms such as spectrogram images, mel-spectrograms etc., are used as input features to DNN models. These models aim to learn higher-level feature representations through a hierarchy of intermediate representations generated from input spectrograms of acoustic events. Mel-spectrogram is a matrix that contains mel band energy values obtained by applying the mel filterbank to the magnitude spectrogram of the signal, frame-by-frame. Further, the log is used to compress the dynamic range of the mel spectrogram, resulting in the log mel spectrogram. Discrete Cosine Transform (DCT) is applied over the log mel spectrogram to obtain MFCCs. Hence computation steps of log mel spectrogram are same as MFCCs till DCT step (Davis and Mermelstein, 1980). The combination of log mel spectrograms and CNNs are widely used for AEC. However, it is hard to map the specific features learned by DNNs to any known features. DNNs learn effectively when large training dataset is used and may not perform well with limited sized training data.

From Table 2.4, it may be observed that SVM is a widely used statistical model for AEC. SVM supports both linear and non-linear kernel functions. Commonly reported kernel functions for AEC in the literature are the linear (Dennis et al., 2011), Radial Basis Function (RBF) (Sharan and Moir, 2015) and intersection kernels (Pancoast and Akbacak, 2012). Linear kernel SVM is popular, simple and has a low computational cost. However, linear SVM does not consider the non-linear nature of input features. RBF kernel is widely used for various applications and it is mainly used when nature of the input features is not fully known. Alternatively, intersection kernel SVM is used to learn from the histogram (BoAW) features in computer vision and it is evaluated by taking the inner product of feature vectors. It is reported thta, intersection kernel SVM learns from nature of input features and outperforms the linear and RBF kernels SVM for AEC (Pancoast and Akbacak, 2012).

However, Computational complexity of non-linear SVM is $\mathcal{O}(n^3)$ in training and $\mathcal{O}(n)$ in testing, whereas computational complexity of linear SVM is $\mathcal{O}(n)$ in both training and testing, where n is the number of support vectors (Yang et al., 2009)(Wang et al., 2010). Linear SVMs learn better with sparse and more discriminative feature vectors of acoustic events (Yang et al., 2009)(Wang et al., 2010), whereas evaluation of kernel from dense feature vectors is more effective for non-linear SVMs (Pancoast and Akbacak, 2012).

2.3 Polyphonic Acoustic Event Detection: A Review

Polyphonic AED is a machine learning problem, that includes broadly two stages. One is feature extraction or feature representation stage and the other is detection or classification stage. During feature extraction, the acoustic features from audio recordings can be represented as an input feature matrix $X \in \mathbb{R}^{N \times T}$, this denotes N acoustic features are extracted from T frames of audio recordings. The labels are represented as target output matrix $Y \in \mathbb{R}^{C \times T}$, where C denotes the number of acoustic event classes in a dataset. The values of Y are binary values. If i^{th} acoustic event present in the j^{th} frame, then $Y_{i,j}$ is set to 1 otherwise it is set to

0 using reference annotation.

Detection stage is further involves two sub-stages. One is learning and the other is the prediction. During learning, model f maps the input X onto output Y. During prediction, for a given feature vector of frame t, $x_t \in X^N$, $f(x)$ outputs a probability value that the acoustic event is present as $p(y_t|x_t, \theta)$, where θ represents the model parameters. The predicted probabilities are then binarized using a fixed threshold to obtain $\hat{y_t} \in [0, 1]^C$. Model parameters are iteratively updated to reduce the error between $y_t \in Y^C$ and $\hat{y_t}$. Further, actual y_t and predicted $\hat{y_t}$ are compared for performance evaluation.

2.3.1 Features

Features used for polyphonic AED are broadly classified into two types. One is monaural and the other is binaural features. Monaural features are extracted from the single channel of an audio recording, whereas binaural features are extracted from both channels of an audio recording. Overlapped acoustic events can be recognized effectively using binaural features. This is similar to the human beings, those use two ears (two channels) to recognize the sounds present in the surrounding environments. A summary of some of the monaural and binaural features are listed in the Table 2.5.

Log mel band energies or log mel spectrograms are the popular features used in most of the state-of-the-art methods of polyphonic AED. Log mel band energies are computed either from monaural or from binaural channels. The Time Difference of Arrival (TDOA) and dominant frequency features proposed by (Adavanne et al., 2017) are specific binaural features. TDOA features are computed based on how microphones (sound source) are spatially located in binaural scenario. The time difference in the frequency bands of binaural channels is exploited for polyphonic AED. In addition, dominant frequencies of the overlapped acoustic events in the lower frequency region of the log mel spectrogram (100 to 400 Hz) from both channels are considered for polyphonic AED.

Table 2.5: A summery of important features and Deep Neural Network models used for polyphonic Acoustic Event Detection

Sl. No.	DNN Models	Input Features	Dataset	References
01	FFNN	Monaural mel and log mel band energies, MFCCs	TUT-SED 2009	(Cakir et al., 2015)
02	BRNN	Raw audio signals	TUT-SED 2009	(Parascandolo et al., 2016)
03	CNN	Monaural log mel band energies	TUT-SED 2016	(Gorin et al., 2016)
04	CRNN	Monaural log mel band energies	TUT-SED 2009, TUT-SED 2016, TUT-SED Synthetic 2016, CHiME-HOME	(Cakir et al., 2017)
05	CBRNN	Binaural log mel band energies, time difference of arrivals, dominant frequencies	TUT-SED 2009, TUT-SED 2016	(Adavanne et al., 2017)

2.3.2 Classifiers

In the beginning, traditional statistical models such as NMF (Mesaros et al., 2015), GMM-HMM (Heittola et al., 2010)(Mesaros et al., 2010) are reported for polyphonic AED in the literature. Deep learning approaches are the state-of-the-art methods for polyphonic AED. A summary of some of the DNN models is given in the Table 2.5. Feed-Forward Neural Networks (FFNN) (Cakir et al., 2015), Convolutional Neural Networks (CNN) (Cakir et al., 2016)(Gorin et al., 2016), Bidirectional Recurrent Neural Networks (BRNN) (Parascandolo et al., 2016) are successfully applied for polyphonic AED and reported significantly better performance compared to traditional statistical classification methods. BRNN is an extension of RNN that allows training in both positive and negative time direction (Schuster and Paliwal, 1997).

Further, CNN and RNN architectures are combined to form Convolutional Recurrent Neural Network (CRNN) model (Cakir et al., 2017), which took advantages of each architecture and performed better than the systems developed using individual architectures. The CRNN consolidates the properties of CNN, which extracts higher-level shift-invariant features and RNN learns long term temporal information of the audio recordings. The CRNN may be considered as the state-of-the-art for polyphonic AED as the approach has reported the best performance. CRNN architecture is widely used to solve recent research challenges such as Detection and Classification of Acoustic Scenes and Events (DCASE) (Virtanen et al., 2016). However, CRNNs require a large dataset for training; if the dataset is not sufficiently large, then this model encounters problem such as overfitting.

From available literature on input features and DNN models for polyphonic AED, it is clear that still there is a scope to develop different feature representations apart from standard log mel band energies for polyphonic AED. As MFCCs, log mel band energies are sensitive to noisy environments. Current state-of-the-art DNN models may not be suitable for smaller sized datasets. Hence, there is scope to design and develop a DNN model that is suitable even for smaller datasets.

2.4 Acoustic Scene Classification: A Review

ASC systems reported in the literature are broadly classified into two types.

1. Individual ASC systems
2. Joint polyphonic AED and ASC systems (Joint model)

Individual ASC systems work in the same way as monophonic AEC systems do. Temporal-spectral, cepstral (Eronen et al., 2006)(Malkin and Waibel, 2005) and TF features (Rakotomamonjy and Gasso, 2015) are used for ASC. Both statistical and DNN models widely used for this task (Geiger et al., 2013; Bae et al., 2016; Mun et al., 2017).

Joint polyphonic AED and ASC system is a single system that recognizes the events and respective scenes concurrently (Bear et al., 2019). Majority of the works reported in the literature consider the polyphonic AED and ASC systems are two different tasks (Stowell et al., 2015). However, the acoustic scene is recognized based on information of acoustic events present in it. Accurate prediction of acoustic events increases the accuracy of the ASC system (Barchiesi et al., 2015). This is similar to human beings use a prior knowledge on presence of likely acoustic events in a scene to recognize the acoustic scene.

Joint model proposed by (Bear et al., 2019), works in the same way as polyphonic AED system does. Feature representation stage of a joint model is explained below in brief. Acoustic features from audio recordings can be represented as input feature matrix $X \in \mathbb{R}^{N \times T}$, this denotes N acoustic features are extracted from T frames of audio recordings. The labels are represented as target output matrix $Y \in \mathbb{R}^{(C+\hat{C}) \times T}$, where C and $\hat{C}$ denote the number of acoustic event classes and respective scenes respectively. The values of Y are binary values. If i^{th} acoustic event present the j^{th} frame, then $Y_{i,j}$ is set to 1 otherwise it is set to 0. Similarly, if j^{th} frame is a part of k^{th} acoustic scene (audio recording), then $Y_{k,j}$ is set to 1 otherwise it is set to 0 using reference annotation.

Combination of monaural log mel band energies and CRNN is used for joint polyphonic AED and ASC. Since there is only one work reported in the literature on a joint model, there is a lot of scope for the design and development of novel features and models to improve the recognition performance of joint model.

2.5 Research Gaps

Some important research gaps are identified from the above review and are listed below.

- Frame-based speech features may not be suitable for AEC. Hence there is a necessity to develop suitable features that can better capture the acoustic event specific information.

- Real-time acoustic events are normally overlapped with high background noise. Frame-based speech features are sensitive to noise. Fisher vector or BoAW representations of frame-based features may not be suitable for AEC especially in noisy conditions. There is a necessity to identify and represent robust features for AEC in highly noisy environments.

- The features from spectrogram are proved to be robust to noisy backgrounds. Feature extraction from two-dimensional spectrograms of large noisy audio event dataset demands high computational time. There is a necessity to develop a time-efficient method for feature extraction from spectrograms.

- DNN models used for monophonic AEC and polyphonic AED require a larger dataset for training. Design and development of a novel DNN model that efficiently works even for smaller datasets is an important research issue.

- There is a necessity to develop an effective system that performs both event detection and ASC. Majority of papers available in the literature addresses these issues as independent problems.

- A standard input to a polyphonic AED is log mel spectrogram. Different varieties of spectrograms can be explored for evaluating the performance of DNN models.

- Polyphonic acoustic events and respective scenes may be recognized better with features from multi channels. Hence, multi-channel features may be explored for improving the performance of joint AED and ASC system.

- In conclusion, research on acoustic event and scene processing is still in its infancy compared to the other audio related speech tasks. Novel features, classifiers and datasets for different applications can be explored further in this regard.

2.6 Problem Statement

Classification of acoustic scenes from large audio recordings by characterization and identification of various acoustic events present. This problem is further elaborated into the following objectives.

1. Characterization and recognition of monophonic acoustic events.

2. Characterization and recognition of polyphonic acoustic events.

3. Recognition of acoustic scenes based on the presence of acoustic events in the audio file.

The defined research problem is elaborated with little insights below. The first objective aims to identify robust acoustic event specific features for classification of monophonic acoustic events in clean and noisy environments. The second objective is to develop an effective model for detection of polyphonic acoustic events. Aim of a third objective is to develop a useful joint model that recognizes both the acoustic events and scene.

2.7 Common Resources used in this Work

Common resources such as datasets, classifiers and baseline systems used for monophonic AEC, polyphonic AED and ASC are explained below in brief.

2.7.1 Common Resources used for Monophonic Acoustic Event Classification

There are two datasets used for evaluation of proposed monophonic AEC systems. One is UPC-TALP dataset (Temko and Nadeu, 2009), used for robust acoustic event specific feature computation and the other is Mivia audio event dataset (Foggia et al., 2015), used to develop time efficient monophonic AEC system.

A UPC-TALP Dataset

Twelve different isolated meeting room acoustic events, namely: applause (ap), cup jingle (cl), chair moving (cm), cough (co), door slam (ds), key jingle (kj), knock (kn), keyboard typing (kt), laugh (la), phone ring (pr), paper wrapping (pw) and walking sounds of steps (st) are selected for monophonic AEC. Approximately 60 acoustic events per class are recorded using 84 microphones. An array of 64 Mark III microphones, 12 T-shape cluster microphones, 8 table top and omni-directional microphones are used for recording. In this work, only the third channel of Mark III array is considered for evaluation. Acoustic events are trimmed to the length of given annotations and resulting data is divided into five disjoint folds to perform five-fold cross-validation. Each fold has equal number of acoustic event clips per class.

To compare the robustness of the proposed approach, 'speech babble' noise from NOISEX'92 database (Varga and Steeneken, 1993) is added to the acoustic events at 20, 10 and 0dB SNRs. All acoustic event clips are available with 44100 Hz sampling rate.

B Mivia Audio Event Dataset

Three classes of acoustic events of interest namely glass breaking, gunshot and human screaming are considered from Mivia acoustic event dataset for surveillance applications. The acoustic events of interest are overlapped with highly noisy background sounds at 5dB, 10dB, 15dB, 20dB, 25dB and 30dB SNR. The background noise includes both indoor and outdoor noises, such as Gaussian noise, crowded ambiance, whistles, rain, bells, household appliances, vehicles, claps, and applauses. It is observed from the dataset that acoustic event like human screaming is similar to the crowded ambiance. Hence, recognition of acoustic events is more challenging. The dataset contains 396 and 184 continuous audio streams of length about three minutes for training and testing respectively. Each audio stream contains three acoustic events overlapped with background noise and only background noise at specific SNR in sequence. The events are trimmed to the length of given annotation. Out of 6000 events of each class, 4200 are used for training and 1800 for testing making 20 hours training data and 9 hours of testing one. All acoustic events are processed at 32000 Hz sampling rate.

C Classifiers

Linear, intersection and Chi-square kernels on SVM classifiers are used for monophonic AEC system. Linear and intersection kernels are computed as given in (Pancoast and Akbacak, 2012). Chi-square distance kernel is computed for SVM as follows. Chi-square distance between any two normalized feature vectors h_1 and h_2 is given in (2.1).

$$\chi^2(h_1, h_2) = \frac{1}{2}\sum_{i=1}^{M}\frac{(h_{1i} - h_{2i})^2}{h_{1i} + h_{2i}} \tag{2.1}$$

Where M is the number of features. The Chi-square distance of feature vectors is computed using (2.1) in a pairwise manner and then, it is converted into the kernel using (2.2) for SVM classification.

$$K_{\chi^2}(h_1, h_2) = e^{-\alpha\chi^2(h_1,h_2)} \tag{2.2}$$

Where α is a constant scaling factor, which is computed as the mean of Chi-square distance between all training features. Lower the Chi-square distance higher the match between features.

D Baseline systems

Two common baseline systems are used for monophonic AEC system evaluation.

1. Mean and standard deviation representations of 13 MFCCs and their first and second-order derivatives are taken over each frame, resulting in 39 × 2 dimensional feature vector.

2. Mean and standard deviation representations of 13 GTCCs and their first and second-order derivatives are taken over each frame Valero and Alias (2012), resulting in 39×2 dimensional feature vector.

2.7.2 Common Resources used for Polyphonic Acoustic Event Detection

The common datasets used for polyphonic AED is explained below in brief.

A TUT-SED 2016 development dataset

TUT-SED 2016 development dataset is developed and released as a part of the DCASE-2016 challenge. It includes manually annotated 22 real-life recordings from two acoustic scenes namely, home (an indoor scene) and residential area (an outdoor scene). Each audio recording is of 3-5 minutes long, resulting in a total of 78 minutes of audio data. The home scene includes eleven annotated acoustic events spread over ten recordings and the residential area includes seven annotated acoustic events spread over twelve recordings. The acoustic events present in home and residential area scenes are given in Table 2.6. All acoustic event clips are recorded with 44100 Hz sampling rate.

2.7.3 Common Resources used for the Joint Model

The common dataset used in this study for polyphonic AED and ASC is explained below in brief.

A Joint Sound Event and Scene Dataset

Joint sound event and scene dataset includes manually annotated 32 acoustic events from ten acoustic scenes. Each scene includes 300 audio recordings of 30 seconds length, resulting in total of 25 hours of audio data. Polyphony level of acoustic events is 3 in any acoustic scene. That means a maximum of three acoustic events are overlapped in any scene. Number of acoustic events in a scene ranges from 1 to number of events in a scene + 1 × polyphony level. The acoustic events

Table 2.6: Acoustic events present in Home and Residential area scenes of 'TUT-SED 2016' development dataset.

Home		Residential area	
Acoustic event classes	Number of acoustic events	Acoustic event classes	Number of acoustic events
rustling	41	banging	15
snapping	42	bird singing	162
cupboard	27	car passing by	74
dishes	94	children shouting	23
drawer	23	people speaking	41
glass jingling	26	people walking	32
object impact	155	wind blowing	22
people walking	24		
washing dishes	60		
water tap running	37		

present in the ten acoustic scenes are listed in Table 2.7. All acoustic event clips are recorded with 44100 Hz sampling rate.

2.8 Summary

This chapter highlighted the available datasets and critically reviewed the features and classifiers used in monophonic AEC, DNN models in polyphonic AED and little work done in joint AED and ASC model. Database section lists different datasets used for monophonic AEC, polyphonic AED and ASC. Monophonic and polyphonic datasets are developed with different intentions and for different applications. A list of developers (universities), size of datasets in terms of number of recordings, number of acoustic events, number of acoustic scenes, length of datasets along with proper references has been given. The features and classifiers for monophonic AEC are reviewed with their success and failures performance for monophonic AEC. DNN models with their strengths and shortfalls are discussed

Table 2.7: Acoustic events present in ten different acoustic scenes of 'joint sound event and scene' dataset.

Acoustic scenes	Acoustic events	Range of number of acoustic events
bus	clear throat, cough, keys, laughter, phone, speech.	1-21
busy street	bus pass by, door close, footsteps, key lock, knock, laughter, motorbike, speech, running, wind.	1-33
office	chairs moving, door slam, drawer, keys, knock, laughter, switch, phone.	1-27
open air market	bag rustle, bus pass by, cooking, footsteps, footsteps on grass, light rain, money, speech, wind.	1-30
park	bus pass by, birdsong, footsteps on grass, gate, laughter, light rain, phone, push bike, speech, wind.	1-33
quiet street	birdsong, footsteps, key lock, light rain, push bike, wind.	1-21
restaurant	chairs moving, cooking, door close, footsteps, laughter, speech.	1-21
supermarket	bag rustle, checkout beeps, footsteps, money, switch, trolley.	1-21
tube	announcement, bag rustle, footsteps, phone, sliding door close, speech, train.	1-24
tube station	announcement, footsteps, running, sliding door close, speech, train.	1-24

in the context of polyphonic AED review. A single joint AED and ASC system with its future improvements in terms of features and DNN models is highlighted in ASC review. The research gaps are listed and problem statement of the current work is done at the end of the chapter. Details of the common resources used in the current work are also given.

CHAPTER 3

Monophonic Acoustic Event Classification

In this chapter, we present three acoustic event specific features: Spectrogram Features (SFs), Spectrogram Image Features (SIFs), MapReduce based Features (MRFs), extracted from spectrograms for monophonic AEC.

3.1 Monophonic Acoustic Event Classification using Spectrogram Features

In this section, the features are extracted from spectrograms through Singular Value Decomposition (SVD). SVD is a popular linear algebra technique introduced by Beltrami and Jordan in 1870's on square matrices (Klema and Laub, 1980). SVD decomposes the matrix into singular values and vectors. Singular values are real positive numbers. Singular vectors are orthogonal in nature (Van Loan, 1976). SVD identifies the dominant variations in the matrix. Higher singular values and respective vectors of a matrix have significant information about the pattern present in the matrix (Boashash et al., 2015). Hence, SVD can be used for object identification in videos (Cernekova et al., 2003) and images (Shi and Malik, 2000). However, to the best of our knowledge, SVD is not explored for AEC tasks.

The main motivation behind this work is, unlike speech with its phonetic representation, acoustic events are short in duration and have distinct TFRs. Hence, the visual information of spectrograms may produce good features for AEC. The detailed method is explained below in brief.

3.1.1 Spectrogram Feature Extraction using Singular Value Decomposition

The steps involved are listed below.

- Generation of logarithmic spectrograms using Short Time Fourier Transform (STFT).
- Resizing the spectrogram to 50×50 TF matrix to reduce computational complexity.
- Increasing the magnitude of the spectral components of a TF matrix by squaring the samples, which enhances the separation between event and non-event parts of the spectrogram.
- Generation of one-dimensional graph signal of size 1×2500 from two-dimensional TF matrix of size 50×50.
- Generation of Laplacian matrix using weight matrix of a graph signal.
- Decomposition of symmetric Laplacian matrix into singular values and vectors.
- Estimation of a threshold from the second smallest singular vector and it divides the singular vector into two parts. One part extracts the high energy spectral components of an acoustic event from the spectrogram.
- High energy spectral components of an acoustic event are considered as feature vectors to SVM for AEC.

These steps are explained below in brief.

A Logarithmic spectrogram generation

The spectrogram is generated using Short Time Fourier Transform (STFT) formulation given in (3.1) (Oppenheim, 1970).

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

Where $\omega(n)$ is a Hamming window function, $x(n)$ is n^{th} sample of a signal in time domain, $X(k,t)$ is the harmonic of k corresponding to the frequency $f_{\mathrm{k}} = \frac{f_{\mathrm{s}}k}{N}$ for frame t, $f_{\mathrm{s}} = 44100$ Hz is a sampling rate. The window length (N) of 256 samples with 50% overlap is considered for spectrogram generation.

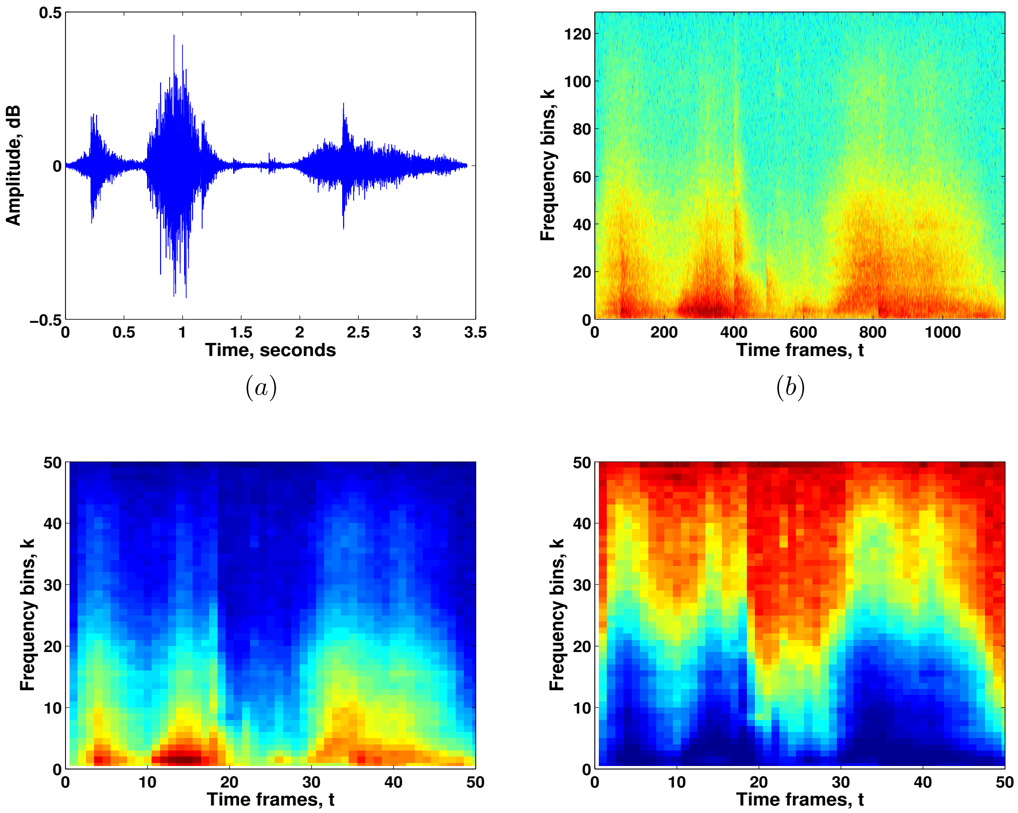
The STFT of an acoustic event gives spectrum with a complex values which includes real and imaginary parts. The magnitude of a STFT yields linear spectrogram and the log is taken to get logarithmic spectrogram using formulations (3.2) and (3.3).

$$S(k,t) = |X(k,t)| \tag{3.2}$$

$$SL(k,t) = log(S(k,t)) \tag{3.3}$$

Where *log* is used to reduce the dynamic range of spectrogram energies and this approach enhances the spectral components belonging to an acoustic event. *SL(k,t)* is a Time-Frequency (TF) matrix, k is a frequency bin of range between 1 to 129 and t is a time frame. Frequency bins $k_{min}=1$ and $k_{max}=129$ corresponding to the zero and 22050 Hz respectively.

Logarithmic spectrogram represents the energy distribution of an acoustic event over time and frequency domains. Majority of the acoustic event energy is concentrated in lower frequencies (see spectrogram of an acoustic event *chair moving* in Figure 3.1*b*). The spectrogram $SL(k,t)$ is resized to 50 × 50 square time-frequency matrix to reduce the computational complexity, at the cost of minimal information loss which is not significant for the work undertaken. The resize operation reduces the resolution of a spectrogram image. However, the high energy spectral components belonging to the event are still unaffected and visible. The logarithmic spectrogram varies along time (t). The resized spectrogram is



(a) (b)

(c) (d)

Figure 3.1: Acoustic event spectrograms. (a) acoustic event *chair moving*; (b) logarithmic spectrogram (SL) of *chair moving*; (c) resized (50×50) spectrogram of *chair moving*; (d) squared values of resized spectrogram of *chair moving*.

invariant to time and resolves the dimensionality ambiguity during feature vector construction. One can differentiate high energy spectral components of an acoustic event in the spectrogram image based on their colors. Blue color corresponds to the low energy, yellow and green correspond to intermediate energy and high energy is reflected through red colored spectral components. High energy spectral components belonging to acoustic events have higher values in $SL(k,t)$, and these are represented by red color in spectrogram image. As the energy of a signal reduces, the spectral components gradually reduce towards low negative values in $SL(k,t)$ and correspondingly color changes from red to yellow, to green and finally to blue in spectrogram images as shown in Figure 3.1c. To enhance the separation of event and non-event spectral components from TF matrix $SL(k,t)$ and to convert logarithmic spectrogram into a graph signal, magnitude of each spectral component is squared as shown in the formulation (3.4).

$$SE(k,t) = (SL(k,t))^2 \tag{3.4}$$

This operation suppresses the magnitude of spectral components belonging to an acoustic event compared to those of non-event. For instance, spectral components of 'plate-sorting' range from -12.535,...,0,...2.0667 in $SL(k,t)$. The smaller positive values from zero onwards belong to the event, smaller negative values from zero downwards belong to the non-event. Squaring of each component transforms the low negative values of non-events to the higher strata compared to the spectral components of the events. Hence, the color of non-event spectral components changes from blue to green, to yellow, to red and color of spectral components belonging to event changes from red to yellow, to green, to blue in spectrogram images as shown in Figure 3.1d.

B Mapping a spectrogram onto a graph

SVD is performed on symmetric Laplacian matrix, which is obtained from the weight matrix of a graph. Hence, one-dimensional graph signal is generated from two-dimensional $SE(k,t)$ matrix as described below.

A graph is defined as $G = (V, W)$, where $V = \{v_1, ..., v_{\mathrm{N}}\}$ is the set of N vertices, W is an undirected weight matrix. First, two-dimensional *SE(k,t)* matrix is converted into a one-dimensional row vector e of size 1×2500 by appending columns of TF matrix $SE(k,t)$ one after the other as given in the formulation (3.5).

$$e = [e_1....e_N]^T \, \epsilon \, \mathbb{R}^N \tag{3.5}$$

The e in (3.5) is a one-dimensional vector known as a graph signal (Sandryhaila and Moura, 2014; Mulimani et al., 2017). The graph signal e is defined from $SE(k,t)$ by mapping set of vertices onto the set of real numbers as shown below.

$$e : V \rightarrow \mathbb{R}$$

$$v_{\mathrm{n}} \rightarrow e_{\mathrm{n}}$$

Where, $\mathbb{R}$ is set of real numbers of length $N = 50 \times 50$

The vector e is normalized to [0-1] scale using the formulation (3.6)

$$e = \frac{e}{max(e)} \tag{3.6}$$

The spectral component e_n of e given in (3.5) is situated at vertex v_{n} in a graph, in other words, e_n is indexed by v_n.

C Weight matrix $W_{\mathrm{i,j}}$

Graph signal is an undirected one. The weight of an edge between i and j is defined by the thresholded Gaussian kernel weight function (Shuman et al., 2013) given by (3.7) and (3.8). The weight matrix of a graph signal is computed using (3.9).

$$P_{\mathrm{i,j}} = \begin{cases} exp\left(- \frac{[dist(v_{\mathrm{i}}, v_{\mathrm{j}})]^2}{\theta^2} \right), & \text{if } dist(v_{\mathrm{i}}, v_{\mathrm{j}}) \leq h \\ 1, & i = j \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

$$Q_{\mathrm{i,j}} = \begin{cases} exp\left(- \frac{[dist(e_{\mathrm{i}}, e_{\mathrm{j}})]^2}{\omega^2} \right), & \text{if } dist(e_{\mathrm{i}}, e_{\mathrm{j}}) \leq h \\ 1, & i = j \\ 0, & \text{otherwise} \end{cases} \tag{3.8}$$

$$W_{\mathrm{i,j}} = P_{\mathrm{i,j}} \times Q_{\mathrm{i,j}} \tag{3.9}$$

Where $dist(v_{\mathrm{i}}, v_{\mathrm{j}})$ corresponds to the Euclidean distance between vertices in graph signal, v_{i} and v_{j}; $dist(e_{\mathrm{i}}, e_{\mathrm{j}})$ represents the physical distance between the spectral components e_{i} and e_{j} of graph signal e indexed by v_{i} and v_{j}, values of the parameters h, θ and ω are empirically chosen to be $5, 0.1$ and 0.3 respectively. A weight matrix $W_{\mathrm{i,j}} \in \mathbb{R}^{M \times N}$ ($M = 2500, N = 2500$) represents the presence of an undirected edge from v_i to v_j with specific weight that indicates the similarity between e_{i} and e_{j}, the spectral components of a graph signal. The spectral components belonging to event and non-event are dissimilar components connected by an edge with higher weight.

D Degree matrix D

Degree matrix D is a diagonal matrix whose diagonal element d_i is the sum of weights of all edges incident on vertex i (Spielman, 2010) and is computed based on (3.10).

$$D(i,i) = \sum_{j} W(i,j) \tag{3.10}$$

E Laplacian matrix L

Laplacian matrix L is defined (Merris, 1994) as (3.11).

$$L = D - W \tag{3.11}$$

Where D is degree matrix, W is weight matrix.

F Singular Value Decomposition

The symmetric matrix $LP = L + L^T$ is decomposed into singular values and singular vectors using (3.12),

$$LP = U \sum V^T \tag{3.12}$$

Where U and V^T are $M \times N$ orthogonal matrices, $\sum$ is $M \times N$ diagonal matrix which has non negative real numbers $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_N$ as its diagonal elements. The diagonal elements σ_{i} of $\sum$ are called as the singular values, Each column of U and V is known as left singular vector and right singular vector respectively (Golub and Reinsch, 1970; Wall et al., 2003). Each column of U or V is associated with the corresponding singular value. The first column (first singular vector) is associated with the larger singular value (σ_1), whereas last singular vector is associated with the smallest singular value (σ_N).

In this work, left singular vectors are considered for feature extraction from spectrograms. The number of sign transitions from positive to negative (zero crossings) is called as the frequency of singular vectors (Shuman et al., 2013). The number of zero crossings of the singular vectors corresponding to higher singular values is higher compared to the singular vectors corresponding to lower singular

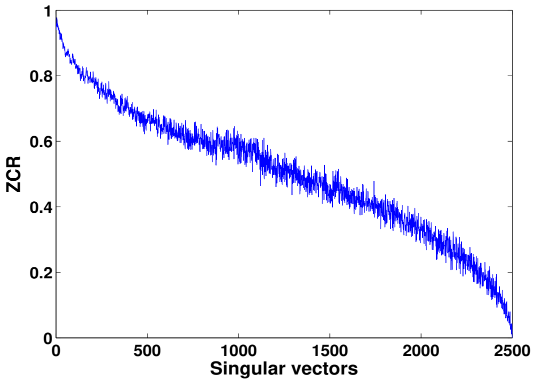
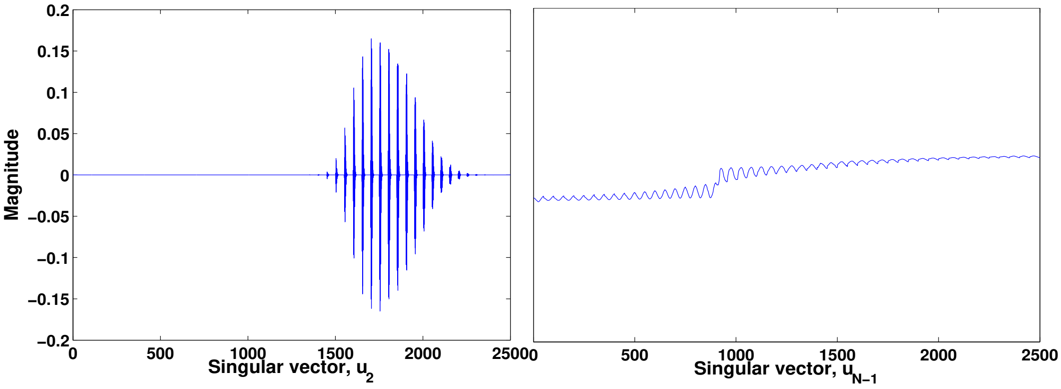


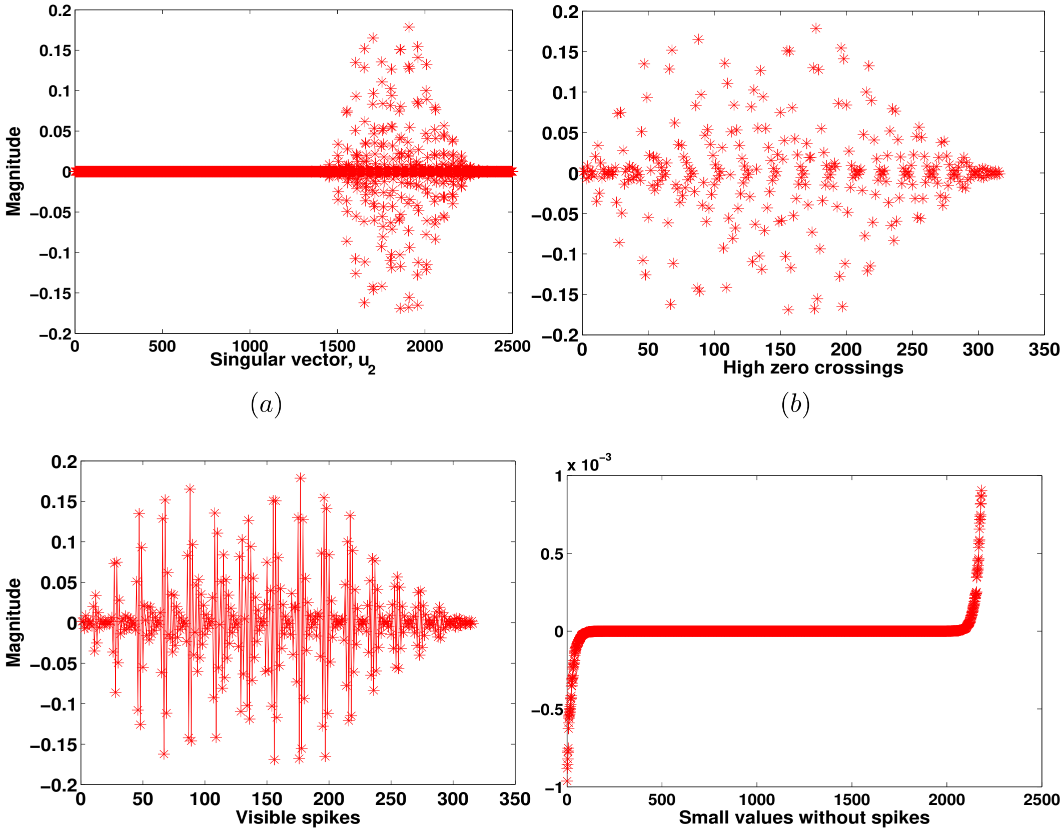
Figure 3.2: Zero crossings in left singular vectors U of *chair moving* after Singular Value Decomposition.



(a) (b)

Figure 3.3: Singular vectors of chair moving obtained after Singular Value Decomposition. (a) singular vector u_1; (b) singular vector u_{N-1} (last but one).

values. For instance, the number of zero crossings of *chair moving* reduces from the first singular vector ($u_1 = 1$) towards higher singular vectors ($u_N = 2500$) gradually (see Figure 3.2). The left singular vectors u_2, u_{N-1} of an acoustic event *chair moving* are shown in Figure 3.3. The singular vector u_N is a constant, and its corresponding singular value is zero (Shuman et al., 2013). As singular values monotonically decrease, and the corresponding singular vectors have similar appearances of sinusoids in the time domain (see $u_N - 1$ in Figure 3.3b). The high positive and corresponding low negative values in u_2 generate visible spikes. These spikes represent the significant information about the energy variation in the signal. The second singular vector u_2 alone effectively characterizes the energy variations in the graph signals due to their unique properties (Mohar et al., 1991;



(*a*) (*b*)

(*c*) (*d*)

Figure 3.4: Acoustic event *chair moving*. (a) singular vector u_2 (b) high zero crossings observed in u_2 ; (c) visible spikes seen in u_2; (d) singular vector u_2 without spikes.

Kim and Mesbahi, 2006). Hence u_2 is considered as the acoustic event specific feature from spectrogram.

G Feature Extraction from Spectrograms using Singular Vector ($\mathbf{u_2}$)

First, we need to identify a threshold E_t that divides the singular vector u_2 of length N into two parts. One part represents the event (high energy spectral components of an event) and the other belongs to the non-event. To perform this, a three-stage approach is proposed. The first one is to preprocess u_2 for identification of threshold E_t, the second one is to define the threshold E_t using the preprocessed singular vector u_2 and the third one is to divide the singular vector u_2 into two parts using chosen threshold E_t. Each stage is explained below in brief.

Preprocessing: By general observation it may be noted that, the values in u_2 are distributed between very high to too small values ranging from $\pm v \times 10^{-01}$ to $\pm v \times 10^{-28}$ in any acoustic event. The values with exponent greater than -03 (such as $-01, -02$) generate visible spikes and their number depends on the duration and energy variations in the signal (shown in Figure 3.4*b*). Clear appearance of visible spikes in u_2 likely represents the events in the spectrogram. Based on this, two empirical assumptions are made. The first one is, the values greater than 10^{-03} and smaller than -10^{-03} generate spikes belonging to the event. These values are identified (shown in Figure 3.4*c*) and excluded from u_2. There are no clear spikes in the remaining values of u_2. The magnitudes of the majority of values are too small, roughly equal to zero (see Figure 3.4*d*) and closely distributed on either side of zero line. Further, we need to identify the positive and corresponding negative values (zero crossings) to understand whether they belong to the event or not. The negative values in u_2 are approximately equal to the corresponding positive values of zero crossings (see y-axis of Figure 3.4*d*). Hence, only negative values are considered. The second assumption is that the values lesser than 10^{-17} and greater than -10^{-17} belonging to the non-event. Hence, too small values near to zero and falling above $-v \times 10^{-17}$ are excluded. Further, it is necessary to analyze the variations in the values from $-v \times 10^{-04}$ to $-v \times 10^{-17}$ and define the threshold, which divides u_2 into two parts namely event and non-event.

Identification of threshold $\mathbf{E_t}$: The absolute values from u_2 with exponents from -04 to -17 obtained from previous preprocessing step are used for threshold identification. The numbers with each exponent (order of magnitude) are named as a sequence. For instance, the sequence with exponent -10 of *chair moving* are given in Table 3.1. It is observed that, generally values v in a sequence, monotonically decrease from nine to one. The values in a sequence beginning with the same value are grouped together and named as the sub-sequence. Few sub-sequences of a sequence with exponent -10 of *chair moving* are given in Table 3.1. Nature of remaining sequences with different exponents remains similar. It is observed that the numbers (v) in the sequences vary by a small margin. Hence,

Table 3.1: Sub-sequences of a sequence of values with exponent -10, chosen from singular vector u_2; Event: *chair moving*

Exponent	Values (v)	Difference $v_{i-1} - v_i$	Remarks
-10	9.8450	-	sub-sequence starts with 9
	9.7721	0.0729	
	$\vdots$	$\vdots$	
	9.3594	-	
	9.0515	0.3079	
	8.5219	0.5296	sub-sequence starts with 8
	8.5087	0.0132	
	$\vdots$	$\vdots$	
	8.1512	-	
	8.0013	0.1499	
	$\vdots$	$\vdots$	$\vdots$
	1.8582	-	sub-sequence starts with 1
	1.6527	0.2055	
	$\vdots$	$\vdots$	
	1.2463	-	
	1.1714	0.0749	

the difference between any two consecutive numbers is less than one. If there are bigger variations in the sequences, then they are the potential points for thresholds for distinguishing the event and non-event. Observation of Table 3.1, 3.2 and 3.3 hint that,

1. Sometimes sub-sequence may not start with nine.
2. Some sub-sequences may be absent.

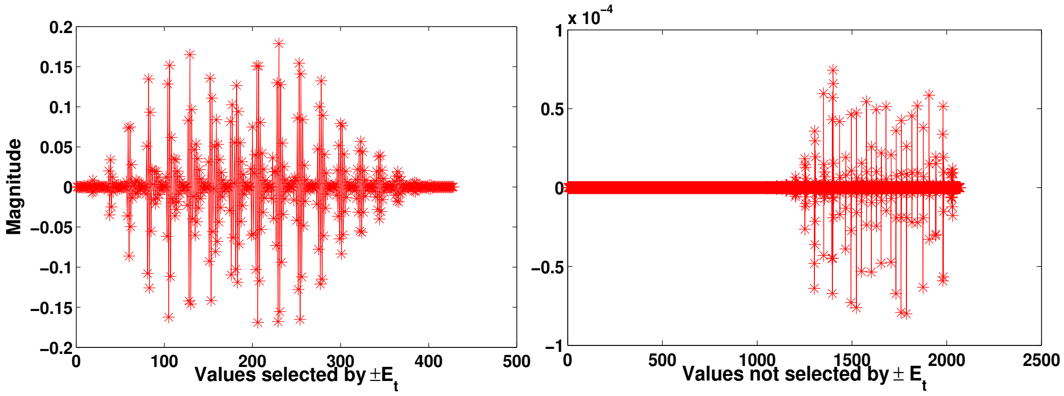
In observation 2, the difference between two particular consecutive numbers in 'u_2' is more than one. For instance, the values in the sequence with exponent -10 of *chair moving* are closely varying and the difference between any two consecutive values (v_{i-1} and v_i) in the sequence is less than one (see Table 3.1). If the difference between v_{i-1} and v_i is less than one, then in this approach, we ignore the change and treat it as no variation. Therefore, there are no variations in the sequence

Table 3.2: Sub-sequences of a sequence of values with exponent -09, chosen from singular vector u_2; Event: *chair moving*

Exponent	Values (v)	Difference $v_{i-1} - v_i$	Remarks
-09	9.2755	-	sub-sequence starts with 9
	9.2629	0.0126	
	⋮	⋮	
	8.8344	-	sub-sequence starts with 8
	8.5386	0.2958	
	8.5168	0.0218	
	8.0145	0.5023	
	6.5388	**1.4757**	sub-sequence starts with 6 Threshold (E_t) is equal to 6.5388×10^{-09} which is the last variation
	⋮	⋮	⋮

Table 3.3: Sub-sequence of a sequence of values with exponent -04, chosen from singular vector u_2; Event: *chair moving*

Exponent	Values (v)	Difference $v_{i-1} - v_i$	Remarks
-04	9.9012	-	sub-sequence starts with 9
	9.1450	0.7562	
	7.9494	**1.1956**	sub-sequence starts with 7 Variation T_h is equal to 7.9494×10^{-04}
	4.9434	**3.0060**	sub-sequence starts with 4 Variation (T_h) is equal to 4.9434×10^{-04}
	2.9197	**2.0237**	sub-sequence starts with 2 Variation (T_h) is equal to 2.9197×10^{-04}
	2.5720	0.3477	
	1.8992	0.6728	sub-sequence starts with 1
	⋮	⋮	



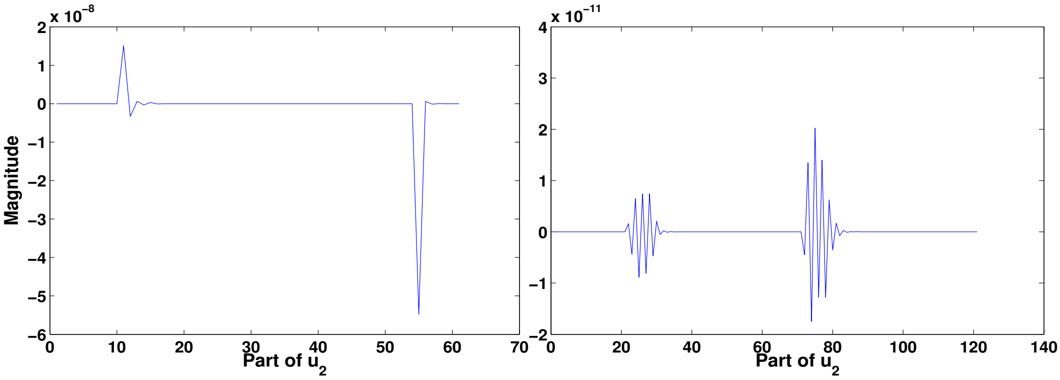
(a) (b)

Figure 3.5: Parts of singular vector u_2 of *chair moving*. (a) values selected by threshold $\pm E_t$; (b) values of u_2 not selected by threshold $\pm E_t$.

with exponent -10. However, in the case of, the sequence with exponent -04, the sub-sequence starting with three, five, six and eight are not present. Hence, the differences between the consecutive v values at this point are higher than one (see Table 3.3) and those are considered as variations (T_h). Similarly, in 'u_2' of some acoustic events, sub-sequence may not begin with nine. This is an instance of the first case. In such situation, the first number of sub-sequence starting with next number is considered as the variation.

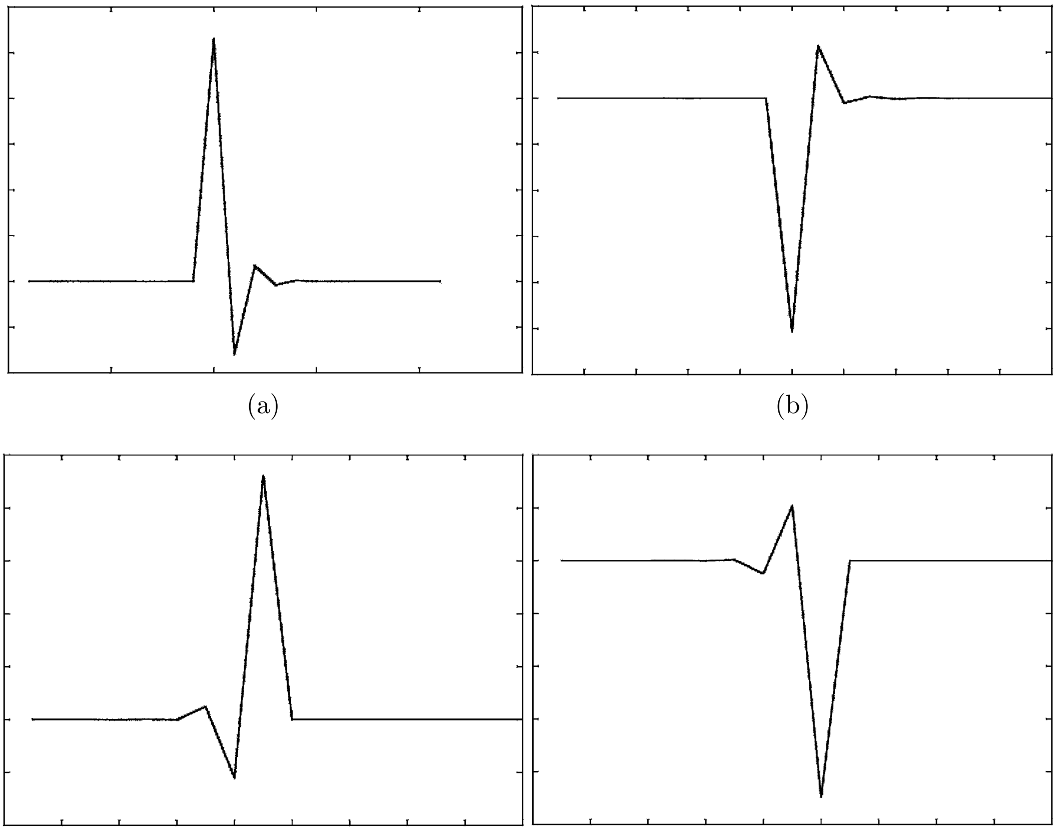
Further, the threshold (E_t) is selected from the list of variations (T_h), that divides u_2 into two parts. One is with variations and other is without variations. For instance, in the case of *chair moving*, last variation is observed in the sequence of exponent -09 (see Table 3.2) and it is considered as the threshold (E_t).

Extraction of features from spectrogram: Threshold E_t which is derived in the previous step, divides u_2 into two parts. One part represents the high-energy spectral components of an acoustic event (spectral shape of an event) in the corresponding spectrogram. The values greater than $+E_t$ and lesser than $-E_t$ of u_2 (in the range $\pm E_t$) belong to one part, and the other values belong to the other part. For instance, the values of u_2 of *chair moving* selected by $\pm E_t$ and values not selected by threshold $\pm E_t$ are shown in Figure 3.5a and 3.5b respectively. The visible spikes are present in both figures (the one, selected by $\pm E_t$ and the other not selected by $\pm E_t$) of u_2. However, the absolute magnitude of



(*a*) (*b*)

Figure 3.6: Types of spikes. (a) Individual spikes observed in singular vector u_2; (b) Continuous multiple spikes observed in singular vector u_2.



(a) (b) (c) (d)

Figure 3.7: Properties of spikes present in u_2. (a) high positive value followed by a low negative extension; (b) high negative value followed by low positive extension; (c) low negative value followed by high positive extension; (d) low positive value followed by high negative extension.

the spikes chosen by $\pm E_t$ is much higher than the spikes not selected by $\pm E_t$ (see y-axis scale difference in the cases Figure 3.5*a* and 3.5*b*). As already discussed, majority of values of u_2 are approximately equal to zero with few exceptions. The high magnitude spikes represent the sudden energy variation in the signal. There are two types of such spikes, namely,

- Well separated individual spikes (see Figure 3.6*a*).
- Continuous spikes (see Figure 3.6*b*).

Further, spikes present in the values of u_2, not selected by the threshold $\pm E_t$, are analyzed. 'u_2' contains both single and continuous spikes. From the preliminary experiments it is observed that, if values of u_2 (not selected by the threshold range $\pm E_t$) contain more number of single spikes with higher positive values followed by lower (absolute) negative extensions (see Figure 3.7a) or higher negative (absolute) values followed by lower positive extensions (Figure 3.7b), like the one shown in Figure 3.6*a*, then part of 'u_2' not selected by the threshold $\pm E_t$ represents non-event spectral components. Part of 'u_2' selected by the threshold $\pm E_t$ represents high-energy spectral components belonging to 'event' of a logarithmic TF matrix SL given in (3.9). If u_2 (not selected by threshold range $\pm E_t$) contains more number of continuous spikes starting with low (absolute) negative values followed by high positive extensions (see Figure 3.7c), or low positive values followed by high negative (absolute) extensions (see Figure 3.7d), like the ones shown in Figure 3.6*b*, then part of 'u_2' not selected by the threshold $\pm E_t$ represents 'event' spectral components. Part of 'u_2' selected by the threshold $\pm E_t$ represents non-event spectral components of a logarithmic TF matrix SL. The large energy variation in a signal generally causes sign transition (either from positive to negative or from negative to positive) in the values of u_2 and the magnitude changes from lower to higher.

High energy spectral components of an acoustic event *chair moving* are extracted using a part of u_2 selected by a threshold $\pm E_t$ (see Figure 3.8*b*1). In the case of *key jingle*, due to frequent and abrupt energy change from low to high,

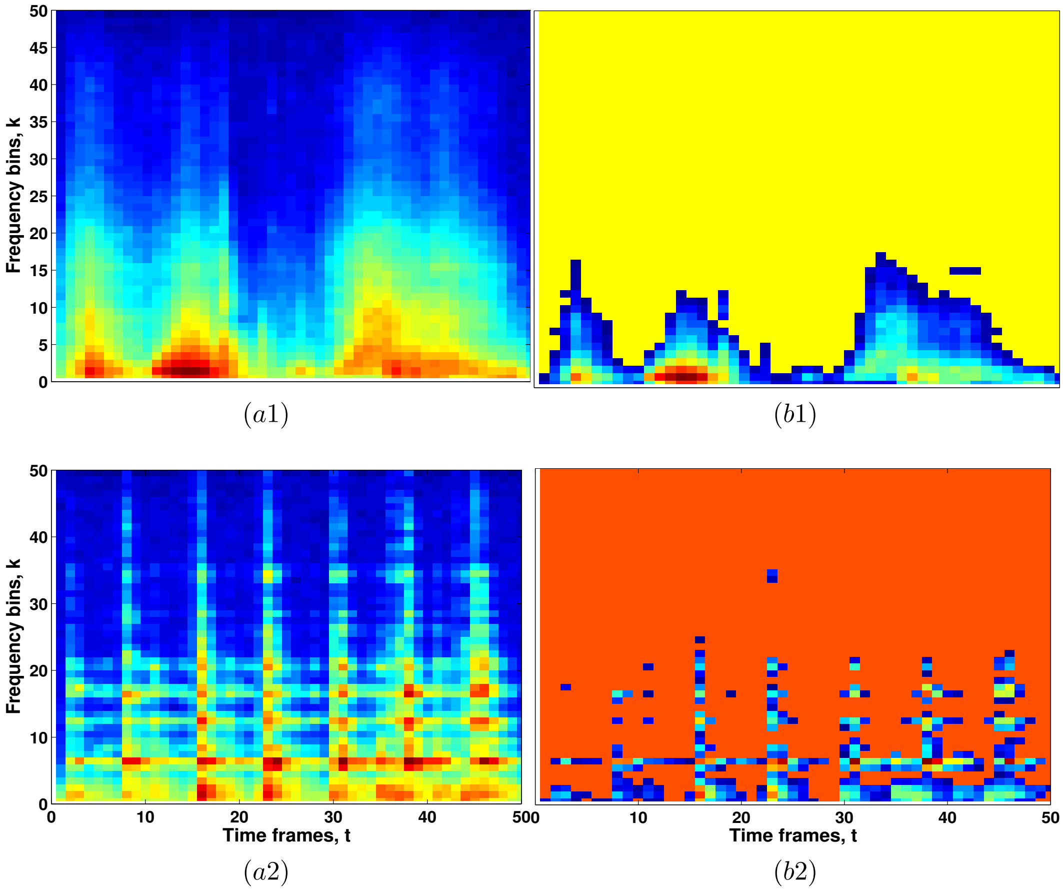


Figure 3.8: Spectrogram feature extraction using singular vector u_2 (a1) logarithmic spectrogram of *chair moving*; (b1) spectrogram features of *chair moving*; (a2) logarithmic spectrogram of *key jingling*; (b2) spectrogram features of *key jingling*.

values of u_2 (not selected by threshold $\pm E_t$) include more number of spikes beginning with lesser value (absolute, Figure 3.7c and Figure 3.7d) and they represent high energy spectral components of a *key jingle* (Figure 3.8b2).

3.1.2 Evaluation

The proposed method is evaluated on UPC-TALP dataset. Recognition accuracy is considered as an evaluation metric. Linear SVM is chosen as a classifier and its optimal parameters are selected using five-fold cross-validation. The proposed method is compared with the method developed using MFCCs as a baseline method.

The high energy spectral components of an acoustic event matrix of size 50×50 (For instance, see Figure 3.8b1 or Figure 3.8b2) is converted into the one-

Table 3.4: Performance comparison of monophonic acoustic event recognition (in %) of the proposed Spectrogram Features with baseline system using SVM classifier.

Features	Accuracy (%)
MFCCs	74.79
Spectrogram Features (SFs)	**80.09**

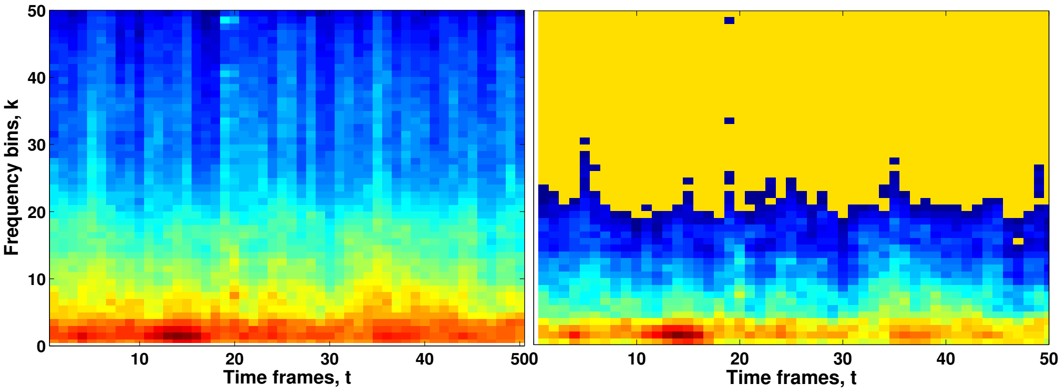
dimensional feature vector Y of size 1 × 2500 by appending its columns one after the other in a row. The third (μ_3) and fourth (μ_4) order central moments are calculated using equation (3.13) from Y and appended to it, that gives 1 × 2502 dimensional feature vector.

$$\mu_{\mathrm{r}} = E(Y - E(Y))^{r} \tag{3.13}$$

where μ_{r} is the r^{th} central moment about the mean μ of the probability distribution Y, E is statistical expectation. High energy spectral components of an acoustic event from a spectrogram and its central moments together are called in this work as a Spectrogram Features (SFs). Further, SFs are normalized between 0 and 1. The average recognition performance of monophonic AEC system with proposed SFs and MFCCs (baseline system) are given in Table 3.4. Results show that proposed SFs recognize acoustic events with an accuracy of 80.09%. The important spectral information of an acoustic event is lost in mean and standard deviation over each frame of MFCCs. Acoustic events have distinct spectral shapes in spectrograms, which are adequately captured by the proposed approach. Hence, the proposed SFs outperform the baseline system through the improvement in marginal.

3.1.3 Contributions and Limitations

To the author's knowledge, this is the first work that has proposed SVD for spectrogram feature extraction. The proposed approach automatically extracts the distinct spectral shape in an acoustic event from a spectrogram without any prior knowledge about the event. The proposed SFs outperform conventional MFCCs and contribute to acoustic event specific features used for AEC tasks.



(a) (b)

Figure 3.9: Acoustic event *Chair moving* with *speech babble* noise at 0dB SNR. (a) logarithmic spectrogram of *chair moving* at 0dB SNR; (b) spectrogram features of *chair moving* at 0dB SNR.

The main limitation of the proposed approach is an issue of sensitivity to the noisy conditions. In a real-time scenario, acoustic events are generally overlapped with high background noise. In such situation, the singular vector fails to discriminate the spectral components belonging to the acoustic event and noise in a spectrogram (see Figure 3.9b, part of a singular vector u_2 selects spectral components belonging to the event and *speech babble* noise at 0dB SNR). Hence, proposed approach may not be suitable during noisy conditions.

3.2 Robust Monophonic Acoustic Event Classification using Spectrogram Image Features

In this work, three different features are evaluated from the spectrogram image for robust AEC; those are listed below and named as Spectrogram Image Features (SIFs).

1. Bag-of-Visual-Words (BoVWs).
2. Fusion Fisher Vector (FFV) Features.
3. Fusion-based Bag-of-Features (FBoFs).

The features extracted from spectrogram images and discussed in the second chapter are robust to noise. However such features are computationally expensive and their statistical representation leads to the considerable loss of information. In this section, different higher-level feature encoding methods such as BoVWs, Fisher vectors are discussed. Generally these are also robust to noise.

BoVWs are widely used in the literature for object recognition in the field of computer vision (Yang et al., 2007). In this work, BoVWs are explored for AEC. Generally, Scale Invariant Feature Transform (SIFT) feature vectors are commonly represented as BoVWs (Lowe, 2004). However, SIFT descriptors effectively recognize objects appear at a different scale, location and poses. An acoustic event in the spectrogram is mostly free from such variations, except variation along time. Hence, SIFT descriptors may not be suitable for AEC. Intensity values of the spectrogram image are considered as features. A combination of BoAWs (representation of speech features) and BoVWs, resulting in FBoFs is also explored for effective AEC.

3.2.1 Robust Acoustic Event Classification using Bag-of-Visual-Words

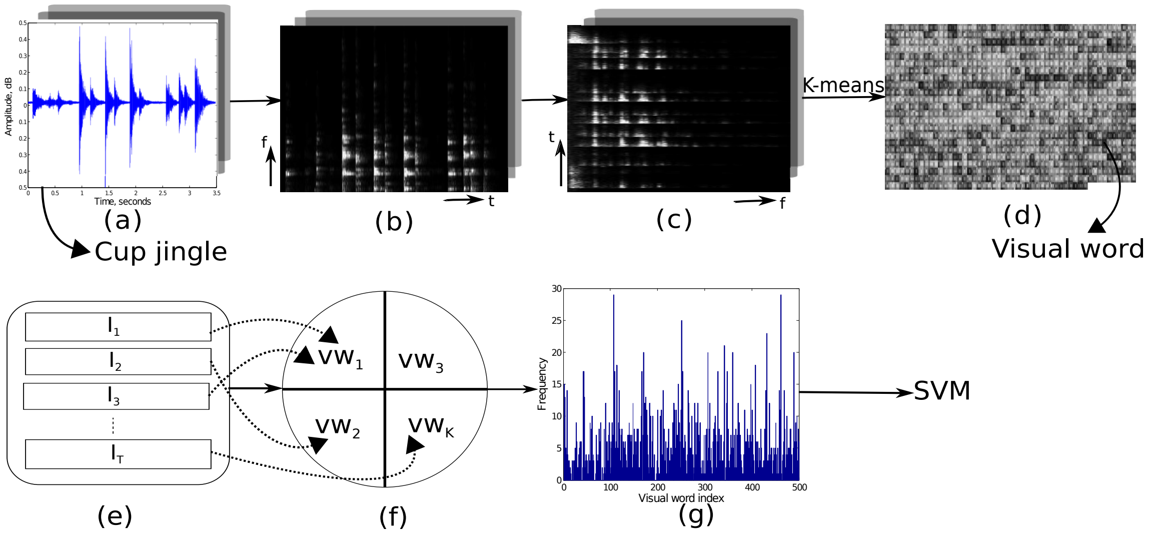


Figure 3.10: Overview of the proposed Bag-of-Visual-Words features. (a) Acoustic events; (b) grayscale spectrogram images of acoustic events; (c) transposed grayscale spectrogram images; (d) visual codebook; (e) and (f) vector quantization; (g) BoVW (histogram) representation

Overview of the proposed approach is given in Figure 3.10. Initially, the grayscale spectrogram is generated from an acoustic event. Visual words are generated from the grayscale spectrogram using k-means clustering. Finally, rows of spectrograms are quantized to get BoVWs as the feature vectors to SVM.

A Grayscale Spectrogram Image Generation

A grayscale intensity spectrogram image is generated (see grayscale spectrogram in Figure 3.10b) by normalizing the values of linear spectrogram $S(k,t)$ (computed using Eq. 3.2) between [0, 1] as given in equation (3.14).

$$GI(k,t) = \frac{S(k,t) - min(S)}{max(S) - min(S)} \tag{3.14}$$

Acoustic events are changing rapidly concerning time, which may cause dimensional variations. Hence, grayscale spectrogram image $GI(k,t)$ is transposed as given in formulation (3.15) to get fixed 129-dimensional row vectors (transposed grayscale spectrogram is shown in Figure 3.10c).

$$G(t,k) = GI(k,t)^T \tag{3.15}$$

Each row of $G(t,k)$ is considered as a 129-dimensional feature vector of intensity values for BoVW representations.

B BoVW Representations

Given a training partition P, containing n randomly selected acoustic events per class represented by $P = p_1, p_2, ..., p_n$, where $p_n = y_1, y_2, ..., y_T$ is the set of feature vectors (rows) of a n^{th} grayscale spectrogram of an acoustic event, where T is the number of time frames. In this work, five grayscale spectrograms per class are randomly selected and those are sufficient enough to generate discriminative BoVWs. Hence $n = 5 \times q$, where q is the number of acoustic event classes. The BoVW model includes two steps: dictionary learning and vector quantization, which are explained below in brief.

Dictionary learning: The K-means clustering algorithm is used to group P into the fixed number of mutually exclusive clusters. The centroids of these clusters are referred to as visual words. All visual words together constitute a vocabulary or a dictionary (shown in Figure 3.10d). There is no known best way to select the size of the vocabulary, i.e., the number of visual words. In this work, the size of the vocabulary ranging from 64 to 512 is considered and its impact on the performance of AEC is analyzed.

Vector quantization: Once the visual vocabulary with M visual words is generated, feature vectors (rows) of a grayscale spectrogram ($G(t, f)$) are quantized to the visual words, i.e., assigned to the nearest visual word in the vocabulary using Euclidean distance (shown in Figure 3.10e and 3.10f). At this point, each feature vector (extracted per time frame) is replaced by the single index, which represents the nearest visual word to that feature vector and this process is known as vector quantization. Further, the histogram or bag (Bag-of-Visual-Words) is generated which gives the number of occurrences of the words in a spectrogram. Finally, normalized (using ℓ_1 normalization) BoVWs are considered as feature vectors to train a classifier.

3.2.2 Robust AEC using Fusion Fisher Vector Features

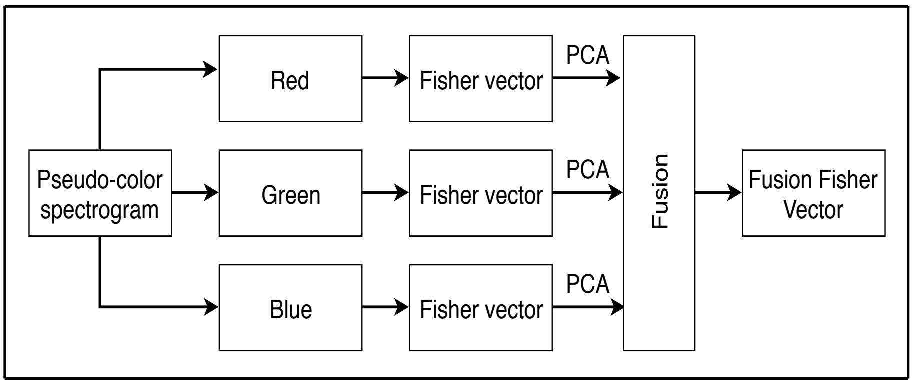
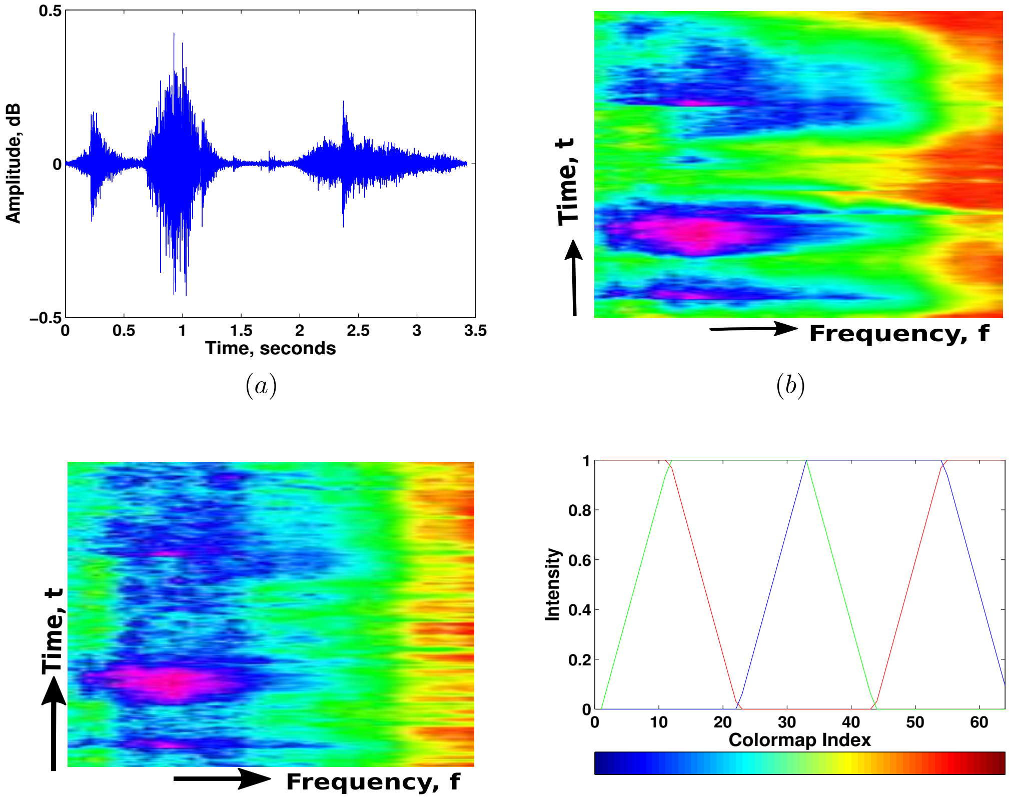


Figure 3.11: Overview of the proposed Fusion Fisher Vector feature extraction.

After generating the pseudo-color spectrogram of an acoustic event, monochrome images are obtained from it and represented them as Fisher vectors. Application of PCA removes the irrelevant features from the Fisher vectors. These are fused to get FFV features later.



(a) (b)

(c) (d)

Figure 3.12: HSV pseudo-color mapped spectrograms of an acoustic event at clean and 0dB SNR. (a) acoustic event chair moving; (b) pseudo-color spectrogram at clean condition; (c) pseudo-color spectrogram at 0dB SNR; (d) HSV colormap.

A Pseudo-color spectrogram generation

In this step, first, an acoustic event is represented as Gammatone spectrogram $S(k,t)$, where k (ranging from 1 to F) is the center frequency of the Gammatone filter and t is the time frame obtained by windowing the signal into frames using the hamming window of length 20ms with 50% overlap. The sampling rate is 44100 Hz and $F = 64$. Along time axis, filters are equally spaced on the Equivalent Rectangular Bandwidth (ERB) scale. The logarithmic Gammatone spectrogram is obtained from $S(k,t)$ using the equation (3.16).

$$S(k,t) = log(S(k,t)) \tag{3.16}$$

Further, values of Time-Frequency matrix $S(f,t)$ are normalized between [0, 1] using the formulation (3.14) to get grayscale intensity spectrogram image. Grayscale spectrogram image is transposed (3.15) to get fixed 64-dimensional row vectors. Grayscale spectrogram ($G(t,k)$) is quantized and mapped onto different monochrome images (3.17) and this process is called as pseudo-color mapping (Dennis et al., 2011).

$$X_q(t,k) = f_q(G(t,k)) \quad \forall q \in (q_1, ..., q_N) \tag{3.17}$$

where X_q is the Red (R), Green (G) or Blue (B) monochrome image, f is the nonlinear mapping function, q is the quantization region (three regions : R, G, and B). In this work, popular HSV colormap is used to map the intensity values of $G(t,k)$ onto RGB monochrome components, resulting spectrogram image is known as pseudo-color spectrogram image (shown in Figure 3.12b) (Dennis et al., 2011).

B Fisher Vector Representations

In this step, each monochrome image of the pseudo-color mapped spectrogram image is represented as a Fisher vector (see Figure 3.11).

Let $X = \{x_t, t = 1, 2, ..., T\}$ be the set of 64-dimensional T row vectors (feature descriptors) of a monochrome image. Where T is the number of time frames. Generally, Fisher vector is derived from the Fisher kernel. The process of generation of Fisher vector includes two stages, the first one is to build the generative model of local descriptors and then obtain feature's coding vector (Fisher vector) by computing gradients of the likelihood of local descriptors concerning the model parameters. In this work, the generative model is Gaussian Mixture Model (GMM), which is trained using the local descriptors of five randomly selected monochrome images per class (small training partition). It is to be noted that, five randomly selected monochrome images per class are sufficient enough to train GMM within a short time. The set of parameters of the trained GMM are denoted as λ :

$$\lambda = \{w_j, \mu_j, \textstyle\sum_j\}_{j=1}^{K} \tag{3.18}$$

Where w_j, μ_j, $\sum_j$ are the weight, mean vector and covariance matrix respectively of a Gaussian j. K is the total number of Gaussians. Each Gaussian is also known as a visual word. All visual words together constitute the visual vocabulary. We assume that $\sum_j$ is a diagonal matrix and diagonal of $\sum_j$ is denoted by σ_j^2, a variance vector of the Gaussian j.

Once the GMM is trained, a monochrome image X is represented as a Fisher vector by assigning its row vector x_t, to the Gaussians.

Let $\gamma_t(j)$ is the soft assignment of x_t to Gaussian j, then

$$\gamma_t(j) = \frac{\exp\left[-\frac{1}{2}(x_t-\mu_j)^T\Sigma_j^{-1}(x_t-\mu_j)\right]}{\sum_{i=1}^{K}\exp\left[-\frac{1}{2}(x_t-\mu_i)^T\Sigma_j^{-1}(x_t-\mu_i)\right]} \tag{3.19}$$

$\gamma_t(j)$ is also known as posterior probability of the Gaussian j (Sánchez et al., 2013).

$V_{\mu,j}^X$ and $V_{\sigma,j}^X$ are the gradients concerning μ_j and σ_j of Gaussian j. They are computed using derivations (3.20) and (3.21).

$$V_{\mu,j}^X = \frac{1}{T\sqrt{w_j}}\sum_{t=1}^{T}\gamma_t(j)\left(\frac{x_t-\mu_j}{\sigma_j}\right) \tag{3.20}$$

$$V_{\sigma,j}^X = \frac{1}{T\sqrt{2w_j}}\sum_{t=1}^{T}\gamma_t(j)\left[\left(\frac{x_t-\mu_j}{\sigma_j}\right)^2-1\right] \tag{3.21}$$

Final Fisher vector V is the concatenation of 64-dimensional gradient vectors $V_{\mu,j}^X$ and $V_{\sigma,j}^X$ of all K Gaussians, resulting in $2 \times D \times K$ dimensional vector. Where D is the dimension of local descriptors, i.e., 64.

In this work, the value of K ranging from 8 to 256 is considered and its impact on the final AEC accuracy is analyzed. Further, Principal Component Analysis (PCA) is applied to reduce the dimension of a Fisher vector from $2 \times 64 \times K$ to M using 'percentage of cumulative variance', which is set to 99% (Jolliffe, 1986). There is no general best practice for selection of the 'percentage of cumulative variance'. 99% of cumulative variance retains the maximum variation among the discriminative features in the M-dimensional Fisher vector with minimal loss of information; hence this approach is chosen for our work. To avoid feature biasing, Fisher vectors are normalized using Signed Square Root (SSR), computed as $V = sign(V)\sqrt{|V|}$, which is also known as power normalization. Further, Fisher vectors are normalized using ℓ_2 norm (Perronnin et al., 2010). Normalized Fisher vectors of three monochrome images of an acoustic event are concatenated (fused) to generate Fusion Fisher Vector (FFV) features.

3.2.3 Robust Acoustic Event Classification using Fusion-based Bag-of-Features

The BoVW representations from grayscale spectrograms obtained in 3.2.1 and BoAW representations from GTCCs of acoustic events are fused to get FBoF representations.

A Gammatone Cepstral Coefficients

The biologically inspired Gammatone Cepstral Coefficients (GTCCs) are derived from the ERB (Equivalent Rectangular Bandwidth) spaced Gammatone filterbanks. The Mel frequency filterbanks are replaced with the Gammatone filters (Valero and Alias, 2012) for obtaining GTCCs. The Gammatone approximation (Unoki and Akagi, 1999) given in (3.22) is used to define the filters in the spectral domain.

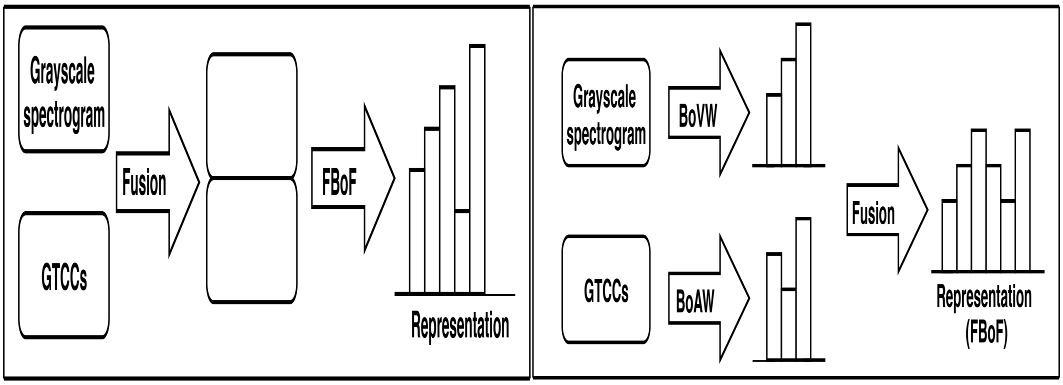
$$M^b(k) = (1 + j(k - k_b)/\omega_b)^{-04} \tag{3.22}$$

Where k_b is the center frequency of the b^{th} Gammatone filter (M), ω_b is the Glasberg Moore bandwidth and j indicates the imaginary unit. The GTCCs and their first and second order derivatives are evaluated over each frame, resulting in a 39-dimensional feature vector for BoAW representation. The BoAW representations of GTCCs are computed in the same way as BoVWs are computed.

B FBoF Representations

The grayscale spectrogram and GTCCs are combined using proposed early or late fusion methods at different levels to obtain FBoFs for AEC.

Early fusion: It is also named as fusion at feature level (shown in Figure 3.13a). A grayscale spectrogram and GTCCs are fused to get $(129 + 39)$ dimensional feature vector per frame for BoF representation. The resulting representation is referred as early FBoF.



(a) (b)

Figure 3.13: Extraction of Fusion-based Bag-of-Features. (a) early fusion at feature level; (b) late fusion at representation level.

Late fusion: It is also named as fusion at the representation level (shown in Figure 3.13b). Each M dimensional BoVW representation from the grayscale spectrogram and BoAW representation from the GTCCs are fused to get $2 \times M$ dimensional late FBoFs.

3.2.4 Evaluation

Performance of proposed spectrogram image features is evaluated on UPC-TALP dataset. Recognition accuracy is considered as an evaluation metric. Linear, intersection, Chi-square SVMs are considered as classifiers. Optimal parameters of SVMs are selected. The proposed methods are compared with the MFCC based baseline method and the following state-of-the-art methods.

1. Pancoast and Akbacak (2012) considered MFCCs and their first and second order derivatives with their log energies are represented as BoAW.

2. Grzeszick et al. (2017) considered combined 13 GTCCs, MFCCs and a loudness over each frame for BoAW representations.

3. SIFs: Concatenating second and third order central moments over 9×9 blocks of monochrome spectrogram images gave Spectrogram Image Features (SIFs) (Dennis et al., 2011).

4. DNNs: Mel band energies are used as features to DNNs, which has three fully

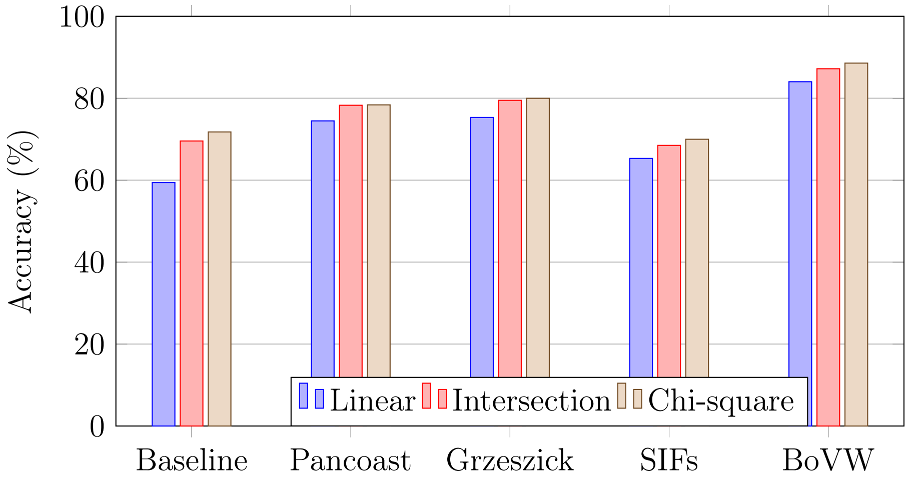


Figure 3.14: Average recognition accuracy of the proposed Bag-of-Visual-Words and other methods using linear, intersection and Chi-square kernels of SVM.

connected layers followed by a softmax output. Each layer uses 500 units with ReLU activation function and 10% dropout. Categorical cross-entropy used as a loss function (Kong et al., 2016).

5. CNNs: 60-dimensional log Mel features are used as input features to Convolutional Neural Networks (CNN). The network had two CNN's of 32, 64 and 128 filters. Each CNN is followed by batch-normalization, Rectified Linear Unit (ReLU) and max pooling. A softmax activation function is used at the output layer (Li et al., 2017).

Performance of the proposed approach is compared with that of the state-of-the-art approaches using UPC-TALP dataset. All considered features are normalized to zero mean and unit variance. The performance comparison of proposed spectrogram features with other methods at clean and noisy conditions on UPC-TALP dataset is given in Table 3.5, 3.6 and 3.8.

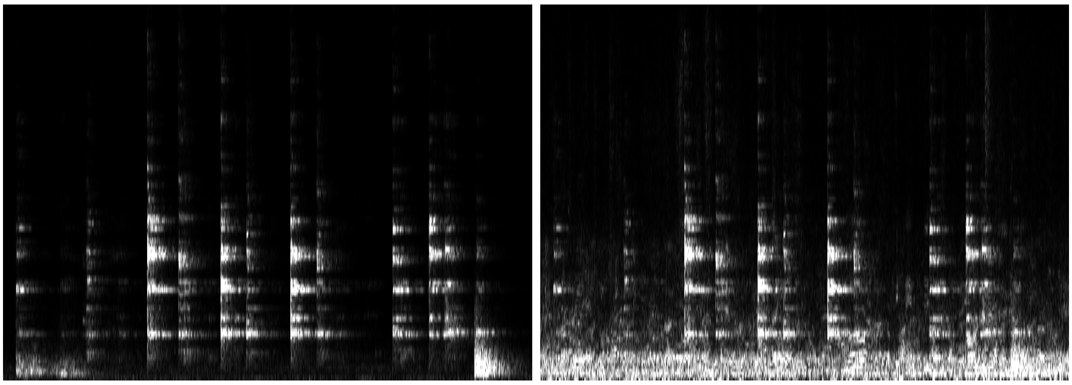
A summary of evaluation results of BoVWs is shown in Figure 3.14. Detailed results at different SNR conditions are given in Table 3.5. The results (Figure 3.1a) show that the Chi-square SVM for AEC, slightly outperforms the Intersection and reasonably outperforms Linear SVM in all the methods. Chi-square and intersection kernel SVMs learn from the nature of input features and achieve high recognition rate, unlike linear SVM. Chi-square and intersection kernel SVMs are

Table 3.5: Performance comparison of monophonic event recognition (in %) of the proposed BoVWs with other methods using Chi-square SVM at clean and different SNR.

Method	Reference	Clean	20dB	10dB	0dB	Average
Baseline	-	83. 76	80.01	71.71	51.67	71.78
Pancoast et al.	(Pancoast and Akbacak, 2012)	88.97	86.88	80.42	57.30	78.39
Grzeszick et al.	(Grzeszick et al., 2017)	89.07	86.97	81.47	59.38	79.22
SIFs	(Dennis et al., 2011)	76.67	75.84	70.17	56.84	69.88
DNNs	(Kong et al., 2016)	70.42	69.38	58.55	35.63	58.49
CNNs	(Li et al., 2017)	86.33	84.05	80.17	65.97	79.13
BoVW	Proposed	**93.54**	**92.51**	**88.76**	**79.54**	**88.58**

commonly used to learn the histogram patterns as features in computer vision. Surprisingly, Chi-square SVM with non-histogram MFCC features outperformed the linear and intersection kernels. Therefore, we consider Chi-square SVM as a competitive classifier in this work. Proposed BoVWs with Chi-square SVM outperform all the existing methods in clean and noise conditions (see Table 3.5). As we had mentioned earlier, the maximum energy of the speech babble noise is concentrated at lower frequencies and MFCCs are sensitive to noise at lower frequencies. Hence, the performance of MFCC based baseline system significantly drops at 0dB SNR. The values of mean and standard deviation computed over each frame lead to the inevitable loss of the useful information. BoAW representations (Pancoast and Akbacak, 2012)(Grzeszick et al., 2017) of frame-based features effectively capture the vital information of the acoustic events and outperform the baseline system. However, it is still inferior to the proposed BoVWs. The magnitude of the spectral components of the acoustic event in the linear spectrogram $S(k,t)$ is much higher than that of the noise. Same phenomenon is observed in grayscale spectrograms as the intensity values of acoustic events are much higher compared to noise. However, noise is commonly more diffuse than the events and maximum energy is spread over lower regions of spectrogram image. Hence, strong peaks of the acoustic events are unaffected by the noise (see Figure 3.15) and are effectively discriminated by BoVW features.

Audio words that are in vocabulary are from the low-level speech features which are sensitive to noise. This reduces the performance of BoAW model at



(*a*) (*b*)

Figure 3.15: Acoustic event cup jingle. (*a*) grayscale spectrogram at clean condition; (*b*) grayscale spectrogram at 0dB SNR.

noisy conditions. Grayscale spectrogram images effectively localize the strongest peaks of acoustic events at 0dB SNR. Hence, visual words of acoustic events are more robust than audio words even in noisy conditions.

Proposed BoVWs fully capture the high-energy intensity values of an acoustic event by considering the entire grayscale spectrogram as a set of feature vectors. SIFs from two central moments over each image block leads to the loss of important information. Hence SIFs are outperformed by the BoVW approach in all conditions.

The performance of the proposed BoVWs using Chi-square SVM is also compared with that of the emerging DNNs and CNNs. The recognition accuracy of the DNNs/CNNs is highly inferior compared to proposed approach. This is because the DNNs/CNNs require huge training data to learn effectively and hence the present dataset is not sufficient for training DNNs/CNNs.

The relative improvement in the recognition accuracy of BoVWs approach with Chi-square SVM vis-a-vis the number of visual words in clean condition is shown in Figure 3.16. One can observe that recognition accuracy improves as there is an increase in the number of visual words. The smaller vocabulary groups the dissimilar acoustic events to the same visual word. Hence, the smaller vocabulary is not discriminative and gives poor performance. On the other hand, the larger vocabulary is more discriminative. However, as the size of the vocabulary increases

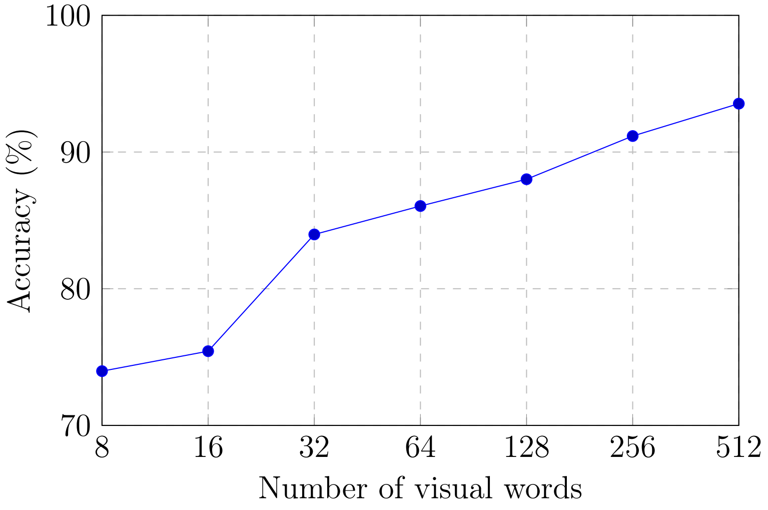


Figure 3.16: Recognition accuracy of proposed Bag-of-Visual-Words for different number of visual words.

Table 3.6: Performance comparison of monophonic event recognition (in %) of the proposed FFV features with other methods using linear SVM at clean and different SNRs.

Method	Reference	Clean	20dB	10dB	0dB	Average
MFCCs	-	74.79	66.46	58.55	46.88	61.67
Pancoast et. al	Pancoast and Akbacak (2012)	88.97	86.88	80.42	57.30	78.39
Grzeszick et. al	Grzeszick et al. (2017)	89.07	86.97	81.47	59.38	79.22
SIFs	Dennis et al. (2011)	75.42	74.01	72.16	55.31	69.22
DNNs	Kong et al. (2016)	70.42	69.38	58.55	35.63	58.49
CNNs	Li et al. (2017)	86.33	84.05	80.17	65.97	79.13
BoVW	Mulimani and Koolagudi (2018)	93.54	92.51	88.76	79.54	88.58
FFV	Proposed	**97.29**	**96.23**	**94.18**	**89.59**	**94.32**

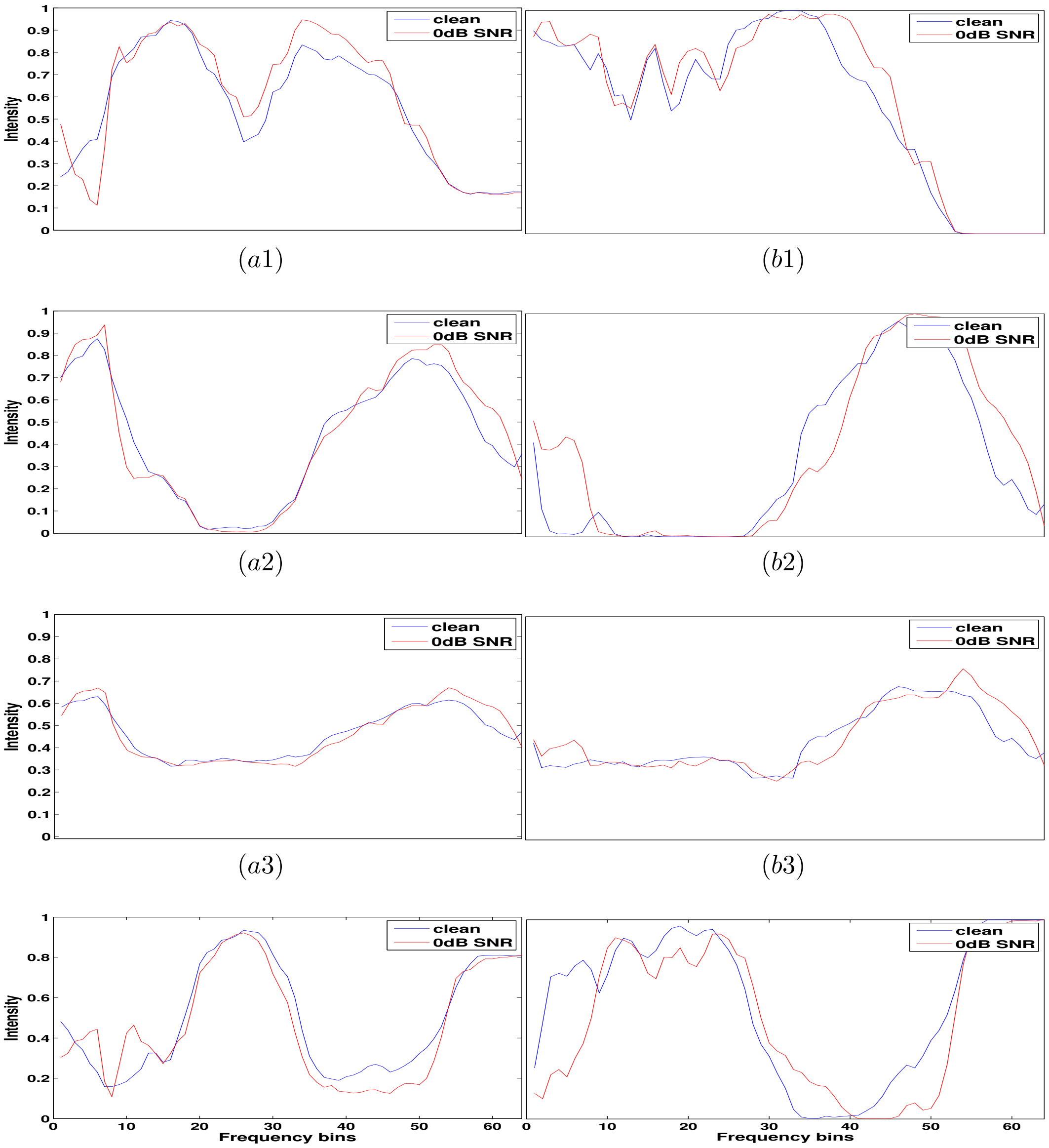
computational complexity also increases. Experimentally we found that 512 visual words (i.e., size of the vocabulary) generate 512-dimensional BoVW representations for AEC with higher recognition accuracy. In this work, results of consistent 512-dimensional representations are reported for all other BoAW experiments and also for proposed BoVWs.

The results given in Table 3.6 demonstrate that proposed combination of FFV-linear SVM outperforms all other known approaches in both clean and noisy conditions with the average recognition accuracy of 94.32%. The proposed FFV features are robust to noise and achieve recognition accuracy of 89.59% at 0dB SNR, which is only 4.73% lesser than average accuracy (94.32%) and 10.05% higher than the next best approach using BoVWs (79.54%).

BoVW representations of the grayscale spectrogram images effectively discriminate acoustic events from the noise and perform better than BoAW representations. However, BoVWs perform well with non-linear kernel classifiers such as Intersection and Chi-square kernel SVMs, which demand higher computational time than the simple linear SVM. Advantages of using the Fisher kernel over BoVW are mainly from two aspects: Fisher vectors can be evaluated from much smaller vocabularies with lower computational time and perform well with linear SVMs (Sánchez et al., 2013). Unlike BoVWs approach, that considers a single nearest visual word for quantization, Fisher vectors consider the probability of event present in each Gaussian (visual word). The dimension of a Fisher vector from each monochrome image is $2 \times D \times K$ while the dimension of a BoVW representation is only K. Hence, the Fisher vector contains significantly much more information about the acoustic events by including gradients concerning mean and standard deviation. Thus, FFV-SVM outperforms the BoVW representations in clean and different noisy conditions.

As we had earlier mentioned that, higher intensity values (stronger peaks) pertaining to the acoustic events are unaffected by noise (see Figure 3.12*c*), which effectively are captured and discriminated by the Fisher vector from the noise and achieve higher recognition accuracy in all conditions compared to other methods.

The Fisher vectors are evaluated from the intensity distribution (normalized spectral energy distribution) in the monochrome images. For instance, the intensity distribution of acoustic events '*chair moving*' and '*laugh*' at clean and 0dB SNR are given in Figure 3.17. In this context, the intensity distribution of a monochrome image is computed as the mean of intensity values (normalized spectral values) of each frequency bin (a total of 64 frequency bins are considered). It is clear evidence from Figure 3.17 that, red, green and blue monochrome images of pseudo-color spectrogram and grayscale spectrogram image (grayscale



($a1$) ($b1$) ($a2$) ($b2$) ($a3$) ($b3$) ($a4$) ($b4$)

Figure 3.17: Intensity distribution of acoustic events chair moving and laugh at clean and 0dB SNR. ($a1$) & ($b1$) spectral energy distribution of red monochrome images; ($a2$) & ($b2$) spectral energy distribution of green monochrome images; ($a3$) & ($b3$) spectral energy distribution of blue monochrome images; ($a4$) & ($b4$) spectral energy distribution of grayscale spectrogram images.

spectrogram is generated using equation (3.14)) of acoustic events have distinct distributions, which are effectively clustered by GMM and quantized by posterior probability. Resulting Fisher vectors recognize the acoustic events with the better recognition rate.

To verify the robustness of individual distribution, Euclidean distance between

Table 3.7: Intensity distribution distances between acoustic events at clean and 0dB noise.

Acoustic Events	Red	Green	Blue	Grayscale
chair moving	0.487	0. 648	0.293	0.659
laugh	0.609	0.835	0.432	0.851

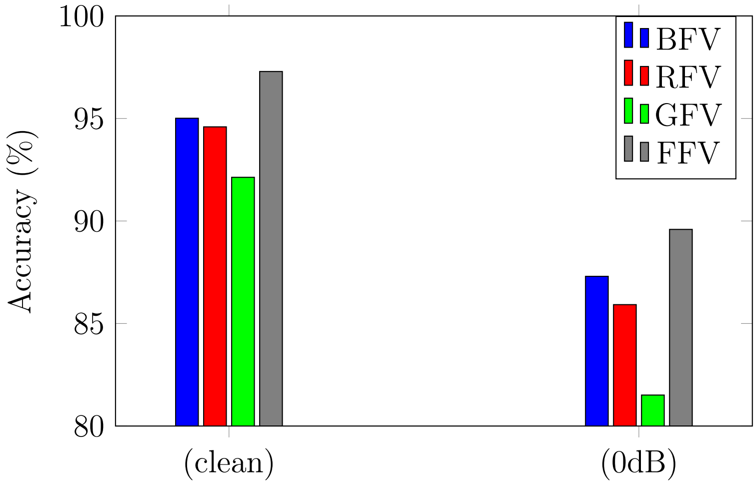


Figure 3.18: Acoustic event recognition accuracy of Red, Green, Blue and Fusion Fisher Vectors at clean and 0dB SNR.

the clean and noisy (0dB SNR) distribution of the same acoustic events (shown in Figure 3.17) are calculated and given in Table 3.7. The similar distributions have smaller distance, indicating that distributions are robust and less affected by the noise. One can observe from Figure 3.17 that, the green and grayscale spectrogram images have more low-intensity values, which are affected by 'speech babble' noise (distributed over the lower regions of spectrograms). Hence, distribution distances of green and grayscale images are high compared to distribution distances of blue and red monochrome images. The higher intensity values have minimal or no effect of noise on them. It is worth to point out that, the intensity distribution of red, green and blue monochrome images of pseudo-color spectrogram are more robust than grayscale spectrogram images. This observation gives us motivation that, Fisher vector representations of red, green and blue monochrome images generate more discriminative and robust FFV features for AEC. Fisher vectors from red, green, and blue monochrome images are evaluated and named as RFV,

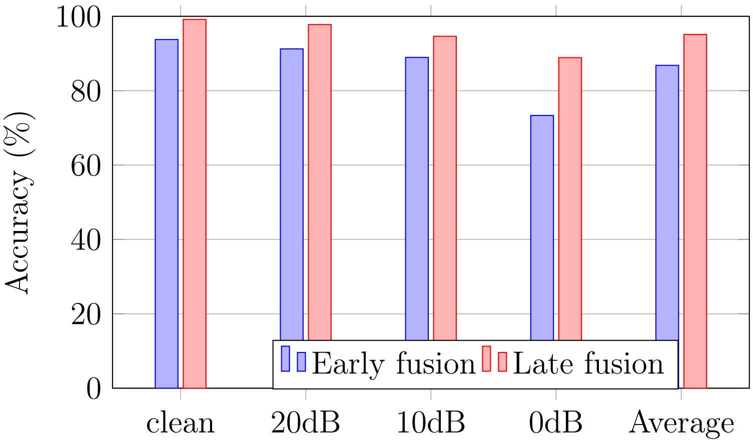


Figure 3.19: Average recognition accuracy of proposed early and late Fusion-based Bag-of-Features at clean and different noisy conditions.

GFV and BFV respectively. A summary of recognition performance of RFV, GFV and BFV along with FFV at clean and 0dB SNR is shown in Figure 3.18. As we had mentioned earlier, Green monochrome image represents the acoustic event in TF dimension with more lower intensity values, which are susceptible to noise as compared to red and blue monochrome images. Hence, RFV and BFV perform significantly better than GFV in both clean and 0dB SNR conditions.

The values with maximum variations in the Fisher vector represent the higher intensity values of the acoustic events in the monochrome images. These significant values are chosen from the Fisher vector using PCA and fused to get FFV. FFV is the combination of selected prominent features from RFV, GFV and BFV. Hence, FFV features exhibit the highest recognition accuracy, which outperforms the Fisher vectors from individual monochrome images, baseline and state-of-the-art methods. The recognition accuracy of Fisher vector representation of grayscale spectrograms is less than BFV and RFV, hence, it is not considered for comparison. Recognition accuracy of FFV-linear SVM concerning the number of Gaussians (K) at clean condition is shown in Figure 3.20. One can observe that recognition accuracy improves by increasing number of Gaussians in the beginning and slowly decreases after 64 Gaussians. Acoustic events are brief and present in the sparse frequency spectrum. Hence, we set number of Gaussians to 64 ($K = 64$), which are efficient enough to recognize acoustic events with a good trade off between accuracy

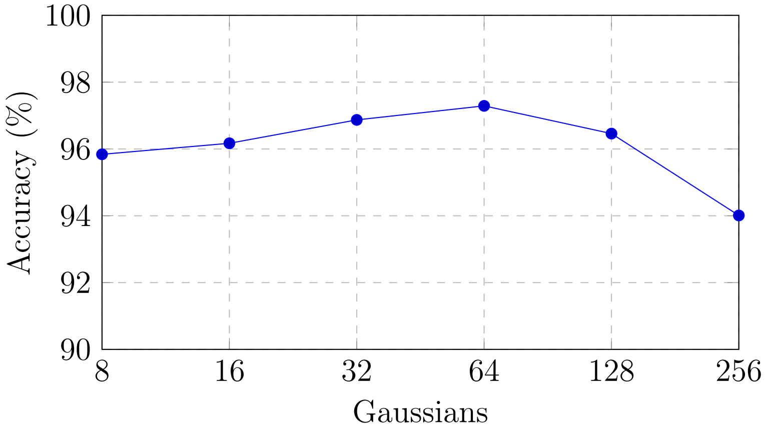


Figure 3.20: Recognition accuracy of Fusion Fisher Vector based features versus number of Gaussians.

and computational cost. Reducing the dimension of PCA may lead to loss of some vital information. Computation of the Fisher vector from each monochrome images of the pseudo-color spectrogram is expensive. Hence, alternative GTCCs are computed from audio recordings using the same Gammatone filterbanks, used to generate Gammatone spectrograms. BoAW representations of GTCCs are fused with BoVWs to get FBoFs for robust AEC.

Performance of the proposed early and late fusion methods to obtain FBoF are shown in Figure 3.19. It may be seen that late fusion outperforms the early fusion in both clean and different noisy conditions. Hence, hereafter, we consider late Fusion-based Bag-of-Features (FBoF) in further experiments. Performance of the proposed FBoF is compared with different approaches at different noisy conditions in Table 3.8. These results demonstrate that proposed FBoFs significantly outperform all other approaches at both clean and noisy conditions.

As we had mentioned earlier, the AEC performance of MFCC baseline system significantly drops at 0dB SNR. Alternatively, GTCCs and their Gammatone filterbank resolution is much higher at lower frequencies with ERB scale than Mel filterbanks with Mel scale. Hence, GTCCs discriminate the spectral components at lower frequencies belonging to the acoustic event and noise more precisely and perform better than MFCC based baseline system.

As expected, BoAW representations of MFCCs and GTCCs outperform the

Table 3.8: Performance comparison of monophonic event recognition (in %) of the proposed FBoF representations using SVM with other methods at clean and varied SNRs.

Method	Reference	Clean	20dB	10dB	0dB	Average
MFCCs	-	83. 76	80.01	71.71	51.67	71.78
GTCCs	-	81. 88	79.42	77.21	65.84	76.08
GTCCs-BoAW	-	92.09	89.38	86.09	77.13	86.17
Pancoast et al.	(Pancoast and Akbacak, 2012)	88.97	86.88	80.42	57.30	78.39
Grzeszick et al.	(Grzeszick et al., 2017)	89.07	86.97	81.47	59.38	79.22
SIFs	(Dennis et al., 2011)	76.67	75.84	70.17	56.84	69.88
DNNs	(Kong et al., 2016)	70.42	69.38	58.55	35.63	58.49
CNNs	(Li et al., 2017)	86.33	84.05	80.17	65.97	79.13
BoVWs	(Mulimani and Koolagudi, 2018)	93.54	92.51	88.76	79.54	88.58
FFV	(Mulimani and Koolagudi, 2019b)	97.29	96.23	94.18	89.59	94.32
FBoFs	Proposed	**99.17**	**97.79**	**94.93**	**89.89**	**95.44**

Table 3.9: Performance comparison of monophonic event recognition (in %) of the contribution of BoVWs with MFCCs and GTCCs using SVM at clean and different SNR on UPC-TALP dataset.

Method	Clean	20dB	10dB	0dB	Average
BoVW+MFCCs	96.46	94.18	89.17	77.92	89.43
BoVW+GTCCs	99.17	97.79	94.93	89.89	95.44

baseline system. The combined GTCCs, MFCCs and loudness as BoAWs (as done in (Grzeszick et al., 2017)) further reduce the performance of GTCCs as BoAWs. Hence, it is not considered further.

Proposed FBoFs are slightly better than FVF features due to information loss during dimensionality reduction of Fisher vectors. However, none of the other methods performed alone as expected. Hence, late fusion method is proposed which concatenates the BoVW representations with GTCC-based BoAWs to get FBoFs. FBoF representations are the combination of features from BoVWs and GTCCs-based BoAWs. They are observed to outperform all other approaches in clean and noisy conditions.

At this point, the fusion of BoVW representation is explored with both MFCC and GTCC based BoAW representations separately. The results are given in Table 3.9. As expected, the combination of MFCC-based BoAW and BoVW representations perform poor, especially during noisy conditions. Hence, GTCCs-

Table 3.10: Performance comparison of monophonic acoustic event recognition (in %) of the proposed FFV features and FBoFs with state-of-the-art method at clean condition on UPC-TALP dataset.

Method	Reference	Accuracy
BoAW - SVM	(Phan et al., 2016b)	96.80
Proposed FFV features - SVM	Proposed approach	**97.29**
Proposed FBoFs - SVM	Improved proposed approach	**99.17**

based BoAW representations are considered as effective BoAWs and fused with BoVWs to get the proposed FBoFs.

The overall recognition accuracy of the FFV features and FBoFs at clean condition is also compared with the state-of-the-art methods on UPC-TALP dataset reported in the literature (see Table 3.9). The traditional speech features are represented as BoAWs (Phan et al., 2016b) and achieve 96.80% of acoustic event recognition accuracy, which is less than using proposed FFV features (i.e., 97.29%) and FBoFs (i.e., 99.17%) in clean condition, on UPC-TALP dataset. However, these speech features are sensitive to noise and performance of their BoAW representations is expected to reduce in noisy conditions.

3.2.5 Contributions and Limitations

BoVWs, FFV features and FBoFs are computed and used as a SIFs for robust AEC. Unlike frame-based speech features, SIFs are computed from visual information of an acoustic event available in a spectrogram. Strongest peaks of an acoustic event in spectrogram are unaffected by the noise; those are effectively identified and used by the proposed SIFs. Results show that SIFs have a significant contribution to robust AEC. Limitations of a proposed SIFs are listed below.

- Rows of a spectrogram as BoVW/Fisher vector representation contain both event and non-event spectral components.
- Representation of an entire spectrogram into a vector of indices of nearest visual words may lead to loss of information.
- Reducing the dimension of PCA may lead to loss of some vital information.

3.3 Extraction of MapReduce-based features from spectrograms for Acoustic Event Classification

In this section, a novel parallel method is proposed for extraction of significant information of the event from spectrogram termed as MapReduce-based features (MRFs), using Google's MapReduce programming model (Dean and Ghemawat, 2008). Extraction of reliable information as features from spectrograms of big noisy audio event dataset demands high computational time. Parallelizing the feature extraction using MapReduce programming model on Hadoop improves the efficiency of the overall system.

The MapReduce model provides the parallel computing environment across the distributed computational nodes of the cluster using Distributed File System (DFS) (Zhang and Chen, 2014). The MapReduce programming paradigm is already implemented in many popular frameworks such as Apache Hadoop with its Hadoop Distributed File System (HDFS) (Shvachko et al., 2010). Hadoop is scalable infrastructure for massive data, which automatically performs data partition, task scheduling and inter-node communication across nodes of the cluster (White, 2012). MapReduce algorithm (job) divides the tasks into the user-defined 'map' and 'reduce' functions. Opensource Python module mrjob (Zhang and Chen, 2014) developed by Yelp is used to implement one or more map and reduce functions (steps) of MapReduce algorithm (job), which is also called as the multi-step MapReduce job. The mrjob runs the steps of MapReduce job on multiple sub-processes of the local system (single system) or on Hadoop cluster, which performs distributed computing on massive data (Manoochehri, 2013). In the current study, the proposed MapReduce job on Hadoop extracts the strongest peaks of the events from the spectrograms in parallel and these are considered as features to train Ensemble Random Forest (ERF) classifier. Further, the runtime of proposed MapReduce job on Hadoop, local system and sequential program for information extraction from spectrograms is compared. Robustness of features from spectrogram are tested in different noisy conditions. The results obtained using the proposed approach are compared with the state-of-the-art methods to establish their significance.

In the recent years, MapReduce model on Hadoop is used for big data processing in different application domains such as data mining (Bhuiyan and Al Hasan, 2015) (Wu et al., 2014), image-video processing (Almeer, 2012)(Heikkinen et al., 2013), social network analysis (Tang et al., 2009) and so on. To the best of our knowledge, this is the first work, that reports the use MapReduce on Hadoop framework for information extraction from large-scale spectrogram TF matrices for AEC, particularly for audio-based surveillance.

3.3.1 Background of MapReduce on Hadoop

MapReduce is a programming model for distributed computation of massive data on different nodes (data nodes) of the cluster. MapReduce algorithm divides the computation problem (job) into two phases namely *Map* and *Reduce*. Each phase contains one or more respective user-specified map and reduce functions, which have a key-value pair as input and output (refer to the general structure of MapReduce algorithm in Figure 3.21). In a <key, value> pair, <value> represents the specific data and <key> uniquely identifies the <value>. 'MapReduce' programming paradigm is implemented in popular opensource framework 'Hadoop'. The overview of MapReduce programming model on Hadoop is shown in Figure 3.22. Hadoop divides the input data (here, spectrograms) into fixed sized disjoint subsets called splits. Hadoop forks one map task (Mapper) for each split which runs the user-specified map function. Likewise, several map tasks run on several nodes of the cluster to process splits in parallel. Time taken to process small splits on each node is much less compared to that on whole data set on a single node. Once all map tasks complete their execution, Hadoop forks reduce tasks (Reducers) on different nodes of the cluster, which are responsible for the execution of reduce function on the intermediate key-value pair from map tasks. Map tasks divide their output key-value pair into partitions. One partition is for one reducer. The values of partitions are sorted by their key and transferred across the network to the nodes where the Reducers are running (White, 2012).

```
/* Map phase                                                    */
method map(key, value)
    Perform computation
    Emit intermediate (key, value)
/* Reduce phase                                                 */
method reduce(key, value)
    Perform computation
    Emit final-output (key, value)
```

Figure 3.21: General structure of MapReduce algorithm.

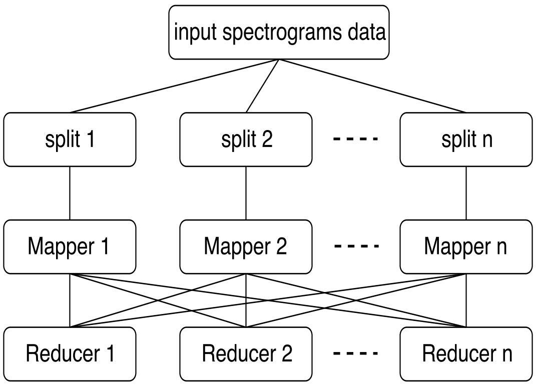


Figure 3.22: Overview of MapReduce programming model on Hadoop framework.

3.3.2 Extraction of MapReduce-based features (MRFs) from spectrogram

In this step, the proposed MRFs extraction from spectrograms for acoustic event recognition is presented. First, a logarithmic spectrogram is generated from an acoustic event. Then, MRFs are extracted from the generated logarithmic spectrogram and used as a feature vector to ERF. The proposed approach characterizes the spectrograms of acoustic events using MapReduce-based features. The algorithm for extracting MRFs from spectrogram is given in Figure 3.23, which is also known as *'MapReduce job'*, or simply a *'job'* and its block diagram is shown in Figure 3.24. The job detects the keypoints (high energy spectral components), which locate the spectral shape or glimpse (Cooke, 2006) of the acoustic event in the spectrogram. Keypoints have rich information about the spectral peaks and ridges in a spectrogram of an acoustic event. The MRFs are obtained from these detected keypoints. Detection of keypoints from spectrogram either row-wise (spectral di-

Algorithm 1: MapReduce-based feature extraction

```
method mapper-one(_ , $row$)
$key \leftarrow row(1)$;
$value \leftarrow row(2 : k + 1)$ ;                               /* $k \leftarrow 129$ */
$fileName \leftarrow get\_filename()$;
emit (($fileName$, integer $key$), real $value$)
end method
method mapper-two($key$, $value$)
$maxima \leftarrow 0, \eta \leftarrow 0, frame \leftarrow 0$
$fileName \leftarrow key[1]$;       /* filename is stored into fileName and
  deleted from $key$ */
for $i \leftarrow 1$ to $k$ do
    for $j \leftarrow -L$ to $L$ do
        if $i + j <= k$ and $i + j > 0$ then
            $block \leftarrow block.append(value(i + j))$
        else
            $block \leftarrow block.append(0)$
    $maxima \leftarrow maxima.append(max(block))$
    $\eta \leftarrow \eta.append(\frac{1}{M} \sum block)$
    $block \leftarrow 0$
for $p \leftarrow 1$ to $k$ do
    for $q \leftarrow 1$ to length(maxima) do
        if $value(p) \geq maxima(q)$ and $value(p) \geq \eta(q)$ then
            $key\_new \leftarrow (fileName, (key, p))$
            $value\_new \leftarrow value(p)$
            emit $(key\_new, value\_new)$
end method
method reducer($key\_new, value\_new$)
$fileName \leftarrow key\_new[1]$; /* filename is stored into fileName and
  deleted from $key\_new$ */
emit $(fileName, (key\_new, unique(value\_new)))$
end method
```

Figure 3.23: Algorithm for identification of keypoints in large-scale logarithmic spectrograms.

mensions) or column-wise (temporal dimensions) using blocks/frames expects high computation time. Parallelizing the keypoint detection using MapReduce would improve the time efficiency. The MapReduce job given in Figure 3.23 (or Figure 3.24) includes two phases namely, *Map* and *Reduce*. *Map* phase contains two steps (methods), namely mapper-one(_, row) which generates the proper <key, value>

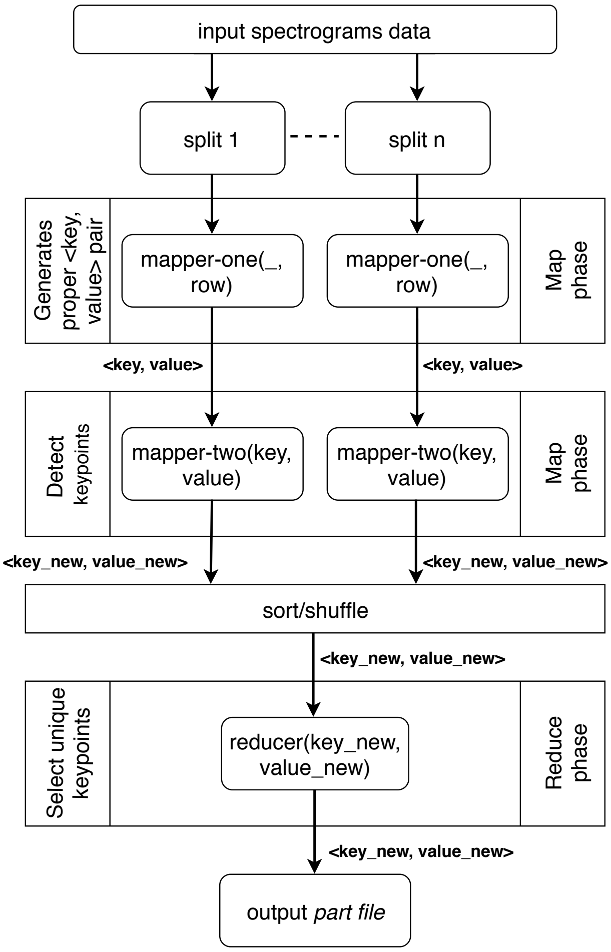


Figure 3.24: Block diagram for identification of keypoints in large-scale logarithmic spectrograms.

```
method local_MapReduce()
foreach spectrogram do
    Generate CSV file
    Execute MapReduce job for keypoint detection (given in Figure 3.4b)
    Write results to local file system
end method
```

Figure 3.25: Local MapReduce algorithm for identification of keypoints in a given spectrogram.

pair from logarithmic spectrogram and mapper-two(key, value), which detects the keypoints in the logarithmic spectrogram. *Reduce* phase has a reducer(key_new, value_new) step, which selects the unique keypoints by eliminating the duplicates.

A Data preparation

Map and reduce tasks accept a <key, value> pair as input and emit the processed <key, value> pair as an output. A row vector of a transposed STFT spectrogram $ST(t,k)$ is considered as <value> and its index (t) is a <key>. A new TF matrix $D(t,k)$ is formed by appending each row vector of $ST(t,k)$ to its index. Hence, the first value of a row in $D(t,k)$ is <key> and remaining values represent <value>. MapReduce system (like Hadoop) operates well on textual data, hence, $D(t,k)$ matrices of the spectrograms are converted into plain-text CSV files.

Further, the CSV data files of the spectrograms are copied to HDFS for distributed computation on Hadoop cluster. Unlike local MapReduce job (shown in Figure 3.25), which extracts the keypoints, spectrogram-by-spectrogram in sequence; Hadoop divides the input spectrograms (CSV data files) into fixed-sized disjoint partitions (subsets) called splits (HDFS blocks). Several splits are distributed among nodes of a cluster for parallel processing. Hadoop forks one map task (on different nodes) to each split which in turn runs the method mapper-one(_, row) of MapReduce job on each row (row-by-row) of input splits in parallel (see block diagram of MapReduce job in Figure 3.25).

B Map phase

Map phase includes two steps/methods namely, mapper-one(_, row) and mapper-two(key, value). Method mapper-one(_, row) is yet another data preparation phase, which reads split row-by-row as an input <value> (input <key> is ignored) and generates proper <key, value> pair for next subsequent method. The first value of the input row (index of row or time frame, t) is considered as <key> and remaining values as <value> to mapper-two(key, value). Further, the pathname of CSV file is combined with the <key> (index of row), which helps to keep track of rows of splits in the distributed environment (refer Line 1 to 6 of algorithm given in Figure 3.23). Finally, <key, value> pair is emitted as an input to the mapper-two(key, value) (see output of mapper-one(_, row) in Figure 3.24).

In mapper-two(key, value), <value> represents the 129-dimensional row vector of $ST(t,k)$ and <key> is its index (t, time frame) with pathname. A row vector <value> is divided into rectangular blocks as $block(q) = value(k \pm j)$, $j = [1, ..., L]$ for keypoint detection. Where $k = [1, ..., 129]$ is a frequency bin, $q = [1, ..., 2L+1]$ is index of a block. Empirically, the value of L is decided to be 3, which is sufficient enough to detect keypoints in <value>. From each block, first, the local maxima is evaluated and appended to a list '$maxima$' as $maxima(b) = max(block)$, $b = [1, ..., NB]$, where NB is the number of blocks and then, noise is estimated and appended to a list 'η' by assuming the noise is stationary across the block (refer Line 7 to 21 of the algorithm given in Figure 3.23) as given in the formulation (3.23).

$$\eta(b) = \frac{1}{M} \sum_{q=1}^{2L+1} block(q) \tag{3.23}$$

In this work, different values of M are considered from 10 to 80 and their impact on the recognition rate is analysed. Experimentally, we found that $M = 40$ recognizes the acoustic events in the noisy environments more with better accuracy. It is worth to note here that, the value of M is independent of the SNR. A significant information from <value_new> is selected from <value> as $value\_new = value(k), \quad \text{if } value(k) \geq maxima \;\&\; \eta$ (refer Line 22 to 30 of the algorithm given in Figure 3.23). A <value_new> is a keypoint which represents the strongest spectral component of an acoustic event in a logarithmic spectrogram. Further, <value_new> and its indices with CSV pathname as <key_new> emitted to reducer(key_new, value_new) as input (see output of mapper-two(key, value) in Figure 3.24). A <key_new> consists of CSV pathname, row (t, time frame) and column (k, frequency bin) indices of interested keypoint <value_new>.

C Reduce phase

A proposed rectangular block moves over a <value> (129-dimensional row vector) in a single step. Hence, the mapper-two(key, value) may emit the same keypoint (<value_new>) with a unique key (<key_new>) more than once. At this point, *Map* phase completes its execution on different input splits of all spectrograms and generates intermediate output. MapReduce framework shuffles and sorts the intermediate keypoints (<value_new>) by its unique key (indices of keypoint) before being sent to the reducer(key_new, value_new). By default, Hadoop forks single reducer task on a node in a cluster after completion of *Map* phase, which runs the reducer(key_new, value_new) method on sorted <key_new, value_new> pair (see *Reduce* phase in Figure 3.24). The reducer(key_new, value_new) outputs the single unique keypoint and its key by discarding duplicates of a keypoint (Line 32 to 35 of the algorithm given in Figure 3.23.

A single reducer task writes the final output (key-value pairs) of the reducer(key_new, value_new) into a file, named as *'part file'* (default output format of Hadoop, see output of reducer(key_new, value_new) in Figure 3.24). *Part file* includes the significant keypoints and their indices (time frame t and frequency bin k) from all spectrograms (input splits) of different events.

Identification of a key-value pair from a particular spectrogram of an event, among all key-value pairs in *part file* is a tedious task. The question may arise that which key-value pair belongs to which spectrogram. To keep track of data, mrjob accepts different output formats from various external libraries with Hadoop. One such library is *NickNack* library used with reducer task. Unlike writing all key-value pair into a single *part file*, *NickNack* writes one *part file* for each spectrogram into HDFS using CSV pathname, which is emitted from *Map* phase.

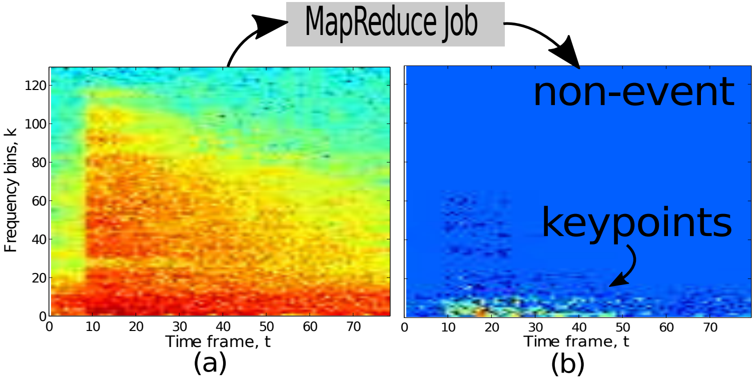


Figure 3.26: Keypoint extraction using MapReduce job. (a) logarithmic spectrogram of a gunshot, which is input to the MapReduce job; (b) identified keypoints as output from MapReduce job.

D Feature vector construction

Part files of respective spectrograms of events are copied from HDFS to the local file system in the form of text files. Each line in a text file composes indices of time frame (t) and frequency bins (k), followed by keypoint belonging to the event in the spectrogram. Reliable spectrogram P^r (shown in Figure 3.26b) of the same size of original spectrogram S (shown in Figure 3.26a) is formed using its respective text file as follows:

$$P^r(k,t) = \begin{cases} \textit{real keypoint}, & \text{if } t,\ k \text{ and } \textit{keypoint} \text{ exist in text file} \\ 0, & \text{otherwise} \end{cases} \tag{3.24}$$

The P^r is a Time-Frequency (TF) matrix which includes most of the zeros representing the non-event subspace ((Cooke et al., 2001)(Raj and Stern, 2005)) of the spectrogram S and few sparsely spaced strongest peaks; those represent the spectral structure of the event.

P^r is resized into 50 × 50 TF matrix to avoid dimensionality ambiguity and computational complexity. The linearized 2500 dimensional feature vector, named MapReduce-based Features (MRFs), is obtained from each acoustic event by appending columns of P^r one after the other to form a single row.

3.3.3 Evaluation

In this work, MapReduce programming job is designed for extraction of features from massive audio data. UPC-TALP dataset used in our previous works is too small to use in distributed environments. Hence, in this work, Mivia audio event dataset is used and is sufficiently large for evaluation of proposed MRFs. Mivia dataset is specifically developed for audio-based surveillance application. Chi-square SVM is considered as a classifier, which outperforms linear and intersection kernel SVMs even on non-histogram features.

Performance of the proposed MRFs is evaluated based on the two main metrics, namely, the average recognition rate (accuracy) of acoustic events and False alarm rate (FAR). False alarms are the rate of misclassification of background noise as one of the events of interest. The false alarm occurs due to detection of an acoustic event of interest when only background sound is present. Further, the miss rate and error are also considered as evaluation metrics for performance comparison. Miss rate is the rate of misclassification of acoustic events as background noises. Hence, it is precisely opposite to FAR. Error is the rate of misclassification of one of the acoustic events as the other one.

Two sets of experiments we are performed to evaluate the performance of the proposed MRFs. The first set of experiments compares the run time of proposed MapReduce job on Hadoop, on the local system and sequential program for keypoint detection. The following systems are developed and evaluated.

1. Proposed MapReduce job on Hadoop for keypoint detection: the mrjob runs the proposed MapReduce job on Hadoop cluster with eight nodes (computers). Hadoop divides these eight nodes into a jobtracker (master) and number of tasktrackers (slaves). The jobtracker tracks the progress of MapReduce job by scheduling the tasks (map and reduce) to run on tasktrackers. Tasktrackers run tasks and send progress reports to the jobtracker. This way the jobtracker maintains the overall progress report of the MapReduce job. If the task fails on a particular tasktracker, then jobtracker reschedules it on the other tasktracker. The cluster is equipped with Intel Xeon octo-core dual-processor with 2.6 GHz speed, 64 GB RAM and 64 bit Ubuntu 16.04 operating system. Network interface between nodes is through dual Gbit ethernet. The cluster is configured with mrjob 0.5.10, Hadoop 2.8.3, Python 2.7.5 and Java 1.7.0. Further, HDFS replication factor is set to 3, block size is set to 64 MB, the number of reducers is set to maximum 10 and MapReduce intermediate compression is enabled with Googles Snappy compression/decompression technique (Rattanaopas and Kaewkeeree, 2017).

2. Proposed MapReduce job on a Local system for keypoint detection: mrjob runs the steps on multiple subprocesses locally (local file system) in parallel which mimics the Distributed File System (DFS) of several nodes, which is not a replacement to a Hadoop.

3. Sequential program for keypoint detection: we implement the proposed

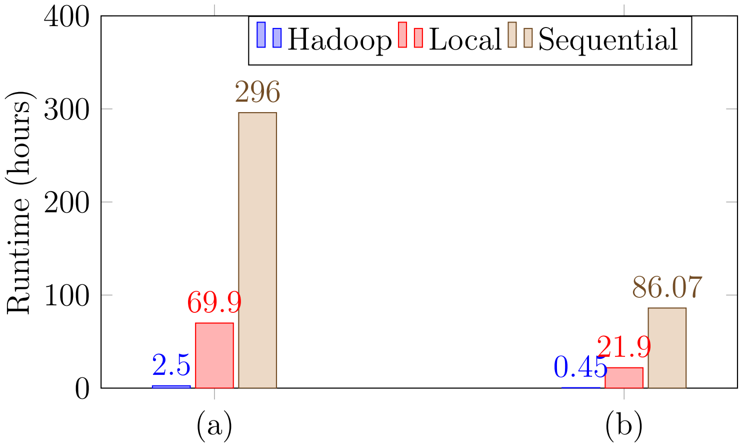


Figure 3.27: Comparison of runtime of MapReduce job on Hadoop and local system with sequential program for keypoint detection. (a) training phase and (b) testing phase using Mivia dataset.

MapReduce job for keypoint detection as a sequential program (see Figure 3.25), which runs on the single system (computer) without mrjob (MapReduce).

The second set of experiments compares the recognition performance of proposed MapReduce based features against state-of-the-art methods.

1. Proposed MRFs and Chi-square SVM classifier.
2. (Conte et al., 2012) use traditional speech features such as spectral, temporal, energy, perceptual features to LVQ (Learning Vector Quantization).
3. (Foggia et al., 2015) use spectral, temporal, energy features as a bag-of-audio-words to SVM.

A Runtime of MapReduce job versus sequential program for keypoint detection

The sequential program runs on a single process and consumes the vast amount of computation time for keypoint detection (see runtime of the sequential program in Figure 3.27 on both training and testing datasets), which is expensive for large-scale audio event datasets. Alternate solution is to use the proposed MapReduce

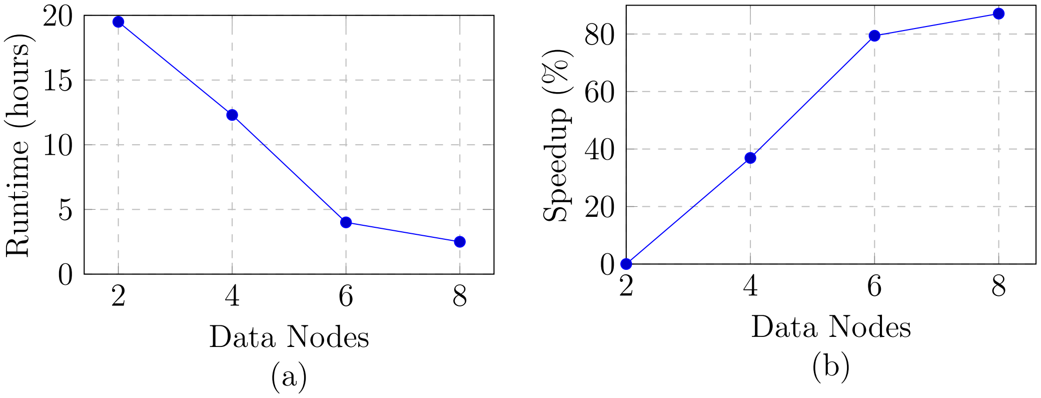


Figure 3.28: Execution time of a MapReduce job with respect to the number of data nodes: (a) line plot shows the runtime concerning number of data nodes; (b) speedup of overall system concerning two data nodes.

job for information extraction from spectrograms (algorithm in Figure 3.23). mrjob runs the steps of MapReduce job locally (on the local file system of a node) and on distributed Hadoop cluster separately. One can observe from Figure 3.27 that, the runtime of MapReduce job on local system drastically reduces as compared to the sequential program. The reason for this is, mrjob runs the steps of MapReduce job on multiple local subprocesses, whereas sequential program runs on a single process. Both sequential program and MapReduce job on local system detect keypoints from spectrograms one by one, unlike Hadoop. Hadoop divides the spectrograms of the whole dataset into 64 MB splits, and automatically schedules the MapReduce tasks on for these splits across nodes of the cluster for keypoint detection in parallel. Hence, the runtime of distributed MapReduce job on Hadoop gets significantly reduced as compared to MapReduce job on the local system and sequential program. Hereafter, we only use MapReduce job on Hadoop with input training data for further analysis of runtime in different conditions.

B Runtime of MapReduce job for varying number of data nodes

Here, we show how runtime of MapReduce job varies with the change in the number of data nodes. For each configuration, the runtime is recorded and shown in Figure 3.28a. We can see that runtime significantly reduces as data nodes increase in number. One can observe from Figure 3.28b that, the overall system speedup sublinearly (almost linear except the last point) increases as more data

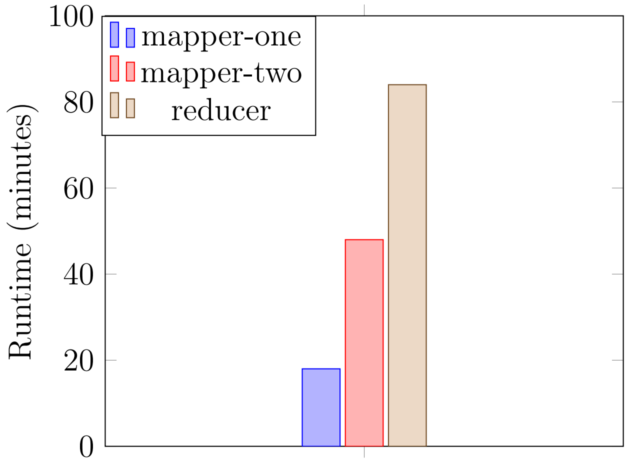


Figure 3.29: Comparison of runtime of map and reduce phases for keypoint detection.

nodes are added. We can observe from Figure 3.28a, after 6^{th} node there is no much reduction in runtime, which is hinted us to have a 8 data nodes are sufficient enough for the given training data set.

C Runtime of MapReduce job for varying number of Reducers

In this experiment, first, runtimes of the map and reduce tasks are analyzed and then, shown how runtime of *Reduce* phase of MapReduce job (given in Figure 3.23) varies with varying number of reducers (reduce tasks). Hadoop forks two Mappers (dual-processor) on each data node (total of eight data nodes), yielding 16 Mappers (map tasks). The mapper-one(_, row) of MapReduce job simply generates proper key-value pairs for mapper-two(key, value). Hence, map tasks take less time to execute mapper-one(_, row) on input splits of spectrograms (see runtime of mapper-one(_, row) in Figure 3.29). The mapper-two (key, value) extracts information from each block over a row of a split (spectrogram) and hence needs high execution time compared to mapper-one(_, row). The processing time of reducer(new_key, new_value) depends on the number of reduce tasks. By default, Hadoop forks single reduce task to a node in a cluster. Reduce task waits until the completion of all map tasks with all input splits. The sorted output of map tasks is transferred over the network to the node, where the reduce task is running. Later output of map tasks are grouped by the key and then fed as

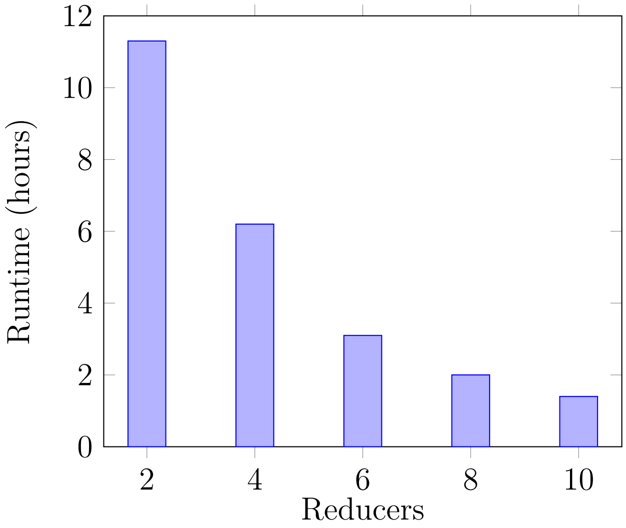


Figure 3.30: Execution time of reduce phase vis-a-vis number of reduce tasks.

Table 3.11: Performance comparison of confusion matrices (in %) for monophonic acoustic event recognition on Mivia dataset using two popular and the proposed MRF-SVM approaches.

	Conte et al. (2012)				Foggia et al. (2015)				Proposed MRF-ERF			
	GB	GS	S	M	GB	GS	S	M	GB	GS	S	M
GB	91.3	5.3	1.4	1.9	94.4	0.2	0.2	5.2	98.0	0.5	0.0	1.5
GS	2.1	80.6	3.9	3.4	3.5	84.9	0.5	11.1	0.0	96.9	0.0	3.1
S	7.6	7.9	79.8	4.7	2.6	0.9	80.8	15.7	0.1	1.1	94.5	4.3

GB: Glass breaking; GS: Gunshot; S:Screaming; M: Miss rate

input to the reducer (new_key, new_value) method. The output of all map tasks to a single reduce task creates burden over the network. The reducer(new_key, new_value) keeps on waiting for the sorted key-value pair of all input splits, then starts which expects more execution time. The solution for this would be to increase the number of reduce tasks. There were 16 reducer slots (processing units) available from eight data nodes. Ten units were used for reducer task by keeping other six units for system use. Where there are more reducer tasks, the map task divides their output into different partitions. Each partition is for a reduce task (number of partitions is equal to the number of reducer tasks), which reduces the burden of single reduce task. Overall runtime of reducer(new_key, new_value) task reduces as the number reduce tasks increases (see Figure 3.30).

Table 3.12: Performance comparison of detailed results of monophonic acoustic event recognition (in %) of two popular and the proposed MRF-SVM approaches at different SNRs.

SNR (dB)	Conte et al. (2012)				Foggia et al. (2015)				Proposed MRF-SVM			
	R	M	E	F	R	M	E	F	R	M	E	F
5	71.4	0.1	28.4	27.9	81.1	12	6.9	11.5	90.3	7.6	2.1	2.2
10	81.2	1.8	17	21.1	85	12.1	2.9	2.4	95.5	4.1	0.4	0.9
15	86.2	3.6	10.2	9.7	87	10.9	2.1	1.3	96.4	3.2	0.4	0.8
20	87.6	4.7	7.8	9.3	88.4	9.9	1.7	1.2	98.4	1.3	0.3	0.7
25	88.2	5.1	6.7	7.2	88.7	9.9	1.4	1.2	98.9	0.8	0.3	0.7
30	88.9	4.9	6.2	7.6	90	9.2	0.8	1	99.3	0.5	0.2	0.7
Avg	83.9	3.4	12.7	13.8	86.7	10.7	2.3	2.6	96.5	2.9	0.6	1

R: Recognition rate; M: Miss rate; E: Error; F: False alarms

D Performance comparison

The recognition performance of the proposed MRF-SVM is compared with the state-of-the-art methods. Confusion matrices of (Conte et al., 2012), (Foggia et al., 2015) and proposed MRF-SVM are shown in the Table 3.11. The detailed results including average recognition rate, miss rate, error, false alarm rates are given in the Table 3.12 at different SNRs. We can observe from the table that bag-of-audio-words with SVM perform well over LVQ. Hence, the approached in (Foggia et al., 2015) accurately detects anomalous events than that mentioned in (Conte et al., 2012). Hence, hereafter, the results reported in (Foggia et al., 2015) are considered as the competitive method for comparison with proposed MRF-SVM. The average recognition rate of the proposed MapReduce-based features with SVM is 96.5%, which outperforms the (Foggia et al., 2015) method. As SNR increases the recognition rate also improves and miss rate reduces. It may be noted that, majority of the errors are due to prediction of the acoustic event as background noise (miss) than misclassification with the other acoustic events (error). As expected, miss rate in the case of human screaming is more than that of the gunshot and glass breaking. This is because person screaming is more similar to the background noise crowded ambiance. However, the magnitude of keypoints belonging to the acoustic events is much higher than that of the noise. The proposed MRF-SVM method effectively discriminates the keypoints belonging to the screaming and other background noises and achieves 94.5% recognition rate with only 4.3% of miss rate, which outperforms the (Foggia et al., 2015). The short acoustic event gunshot is more sensitive to the noise but has unique spectral shape leading to the proper identification of keypoints by the proposed MapReduce job during MRFs extraction. The vector quantization method using low-level features, such as spectral features, energy features and temporal features (Foggia et al., 2015)(Carletti et al., 2013) fails to discriminate the gunshot with background noises; resulting in a low accuracy compared to proposed MRF-SVM. The proposed MRFs are robust and achieve average 90.3% recognition rate at 5 dB noise condition, which is only 6.2% lesser than average recognition rate (96.5%) of the test dataset. Further, the low-level features are sensitive to the noise and generate more false alarms; hence, FAR of (Foggia et al., 2015) is higher in the case 5dB SNR compared to the proposed MRF-ERF method. Overall the proposed MRF-ERF performs well in the cases of wide range of real-time background noises and recognizes the acoustic events with better accuracy.

3.3.4 Contributions and Limitations

To the best of our knowledge, this is the first work, which uses MapReduce on Hadoop framework for information extraction from spectrograms of massive audio data for AEC. Proposed MapReduce job executes on distributed nodes of a cluster simultaneously, resulting in reduced computational time. The proposed MRF-SVM approach achieved 96.5% recognition rate with 1% FAR. It indicates that proposed MRFs have a significant contribution towards the characterization of acoustic events. The results also show that MRFs are robust to noise and achieve 90.3% AEC recognition rate even at 5dB SNR.

MapReduce programming approach is designed for processing a massive audio data. If the size of the data is not sufficiently large, then benefit of using clusters may reduce accordingly. It is because small data creates system overhead (such as initial setup of communication among nodes, task scheduling and so on) and demands high computational time at the cost of actual throughput.

3.4 Summary

In this chapter, SFs, SIFs and MRFs are explored for characterization of monophonic acoustic events. SFs are acoustic event specific. However, they are not robust to noise. Hence, SIFs such as BoVWs, FFV features, FBoFs are explored. FBoFs are found to exhibit a robust acoustic event discriminative characteristics. SIFs are basically computed from spectral components belonging to an event and non-event parts of a spectrogram. Extraction of high-energy spectral components belonging to events from spectrograms of big noisy audio event dataset demands high computational time. Parallelizing the feature extraction task using the MapReduce programming model on Hadoop improves the efficiency of the overall system. A novel parallel method is proposed for the extraction of significant information of the events from spectrogram as MapReduce-based features (MRFs) using Google's MapReduce programming model.

CHAPTER 4

Polyphonic Acoustic Event Detection

In chapter 3, SFs, SIFs and MRFs were extracted from the spectrograms and used to train the SVM model for the monophoic AEC. It recognizes at most one acoustic event at a given instance of time. In a real-time scenario, more than one acoustic events may overlap at any given instance of time. If the task is to detect multiple acoustic events at a particular time then the monophonic AEC model fails. To overcome this drawback of monophonic AEC, polyphonic AED models are presented in this chapter for the detection of overlapped acoustic events. The major advantage of the polyphonic AED model over the monophonic AEC model is as follows. The polyphonic AED system detects both monophonic and polyphonic acoustic events in a continuous audio signal, in contrast to the monophonic AEC system that assigns an audio signal to one of the acoustic event classes. Currently, deep learning-based polyphonic AED models used in the state-of-the-art and reported in the literature. In this chapter, we present two deep learning models; one is Convolutional Recurrent Neural Network (CRNN) and the another is a DNN-driven feature learning approach for polyphonic AED.

4.1 Polyphonic Sound Event Detection using mel IFgram Features and Convolutional Recurrent Neural Network

CRNN may be considered as the state-of-the-art approach for polyphonic AED as the approach has reported the best recognition performance in the literature (Cakir et al., 2017). CRNN captures both spectral (by CNN) and temporal (by RNN) information from acoustic events. In this work, the performance of the CRNN is further improved by using the bidirectional Gated Recurrent Unit (GRU) and mel IFgram (Instantaneous Frequency gram) features as input features for polyphonic AED rather than using common log mel band energies.

The proposed methodology involves two stages. The first one is the extraction of mel IFgram features and the other is the development of the CRNN model for polyphonic SED. Each stage is explained below in brief.

4.1.1 Extraction of mel IFgram features

An audio signal is represented using its sinusoidal components which can be written as a function (4.1).

$$s(t) = r(t) \cdot \cos(\theta(t)). \tag{4.1}$$

where s denotes the signal, $\theta(t)$ denotes the phase, r denotes the amplitude and t denotes the time instance (Abe et al., 1995). Instantaneous Frequency (IF) $\phi(t)$ is defined as the derivative of the phase of an audio signal with respect to time as given in (4.2).

$$\phi(t) = \frac{d\theta(t)}{dt} \tag{4.2}$$

IF features are computed using (4.2) from 40 ms frames with 50% overlap from an acoustic event signal. A dot product between IF features and mel filter banks with specific range of frequencies generate mel IFgram features.

A proposed F-dimensional feature vectors of mel IFgram features, each from T frames in a sequence form a feature matrix $X \in \mathbb{R}^{T\times F}$. Further, feature matrices of the multiple acoustic events normalized using zero mean and unit variance scaling and considered as input data to the CRNN for training.

4.1.2 Architecture of Convolutional Recurrent Neural Network

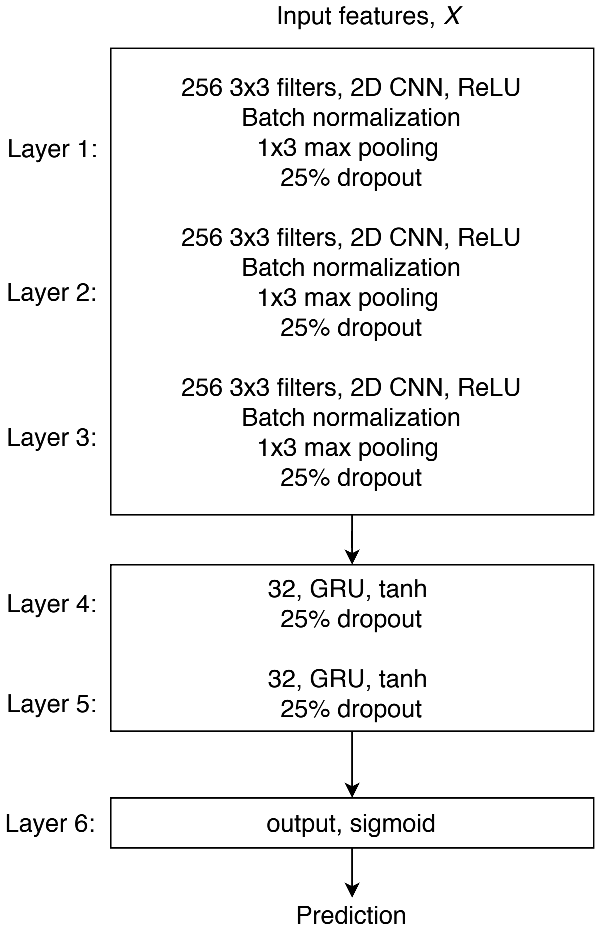


Figure 4.1: Overview of proposed framework of Convolutional Recurrent Neural Network for polyphonic Acoustic Event Detection.

The architecture of the proposed CRNN model is shown in Figure 4.1, it includes three 2D convolution layers, each followed by Relu activation, batch normalization, max pooling and a dropout layer. The output of CNN is fed as input to the first layer of two bidirectional GRU layers. Further, the second bidirectional GRU layer is followed by a fully connected output layer with a sigmoid activation function for polyphonic AED. Each layer of the CRNN is explained below in brief.

A CNN layer

The CNN layer processes a small local region of the input using the group of filters followed by the max-pooling layer. These filters slide over the input feature matrix step by step from left to right. During convolution, some filters may get a stronger response from some local regions of the input feature matrix, while the other regions are suppressed. Hence, convolutional operation effectively captures the high energy spectral components of the acoustic events from the input feature matrix.

Generally, CNN gives the higher-level feature vectors from the input feature matrices, of size defined in equation (4.3) as output.

$$o = (i - k) + 2p + 1 \tag{4.3}$$

where o refers to the output size, i represents the input size and k represents the filter size and p represents the padding of zeros to the input feature matrix X. In other words, convolutional operation downsamples the input feature matrix of size i to o.

In this work, CNN maps the input feature matrix of size i to o using (4.4).

$$o = i - 2n - 1 + 2n + 1 \tag{4.4}$$

where $2n+1$, $n \in \mathbb{N}$, is the k filter size when it is odd and $p = \left\lfloor \frac{k}{2} \right\rfloor = n$ represents the padding, which is also known as half padding. It is to be noted that, size of the output generated by convolution layer is equal to its input size (refer 4.4) and it helps to maintain the alignment between the output of hidden units of CNN and a target vector y.

The feature map of a CNN layer is an output through ReLU (Rectified Linear Unit) activation function, batch normalization (Ioffe and Szegedy, 2015), max-pooling and dropout (Srivastava et al., 2014). Activation functions such as Relu, tanh (tangent hyperbolic), sigmoid and so on are responsible for transforming the dot product between input and filter from a hidden unit of a network into the activation of the unit or output of the unit (Karlik and Olgac, 2011). ReLU activation function is defined as given (4.5).

$$a = max\{0, z\} \tag{4.5}$$

Where z is the dot product between input and filter of a hidden unit, which is also referred as weighted sum. ReLU offers a higher gradient when z is positive than sigmoid activation function (Cakir et al., 2017). Hence, ReLUs are the most commonly used activation functions for the hidden layers of CNN for polyphonic AED.

Batch normalization and dropout are also known as network regularizers, which are explained below in brief. Batch normalization generalizes the unseen data and improves the performance of a CNN. It normalizes the output (activation) of a previous layer using zero mean and unit standard deviation. Dropout reduces the overfitting by ignoring the output of randomly selected hidden units with a specific probability. Hence, there is no significant effect of ignored hidden units on the output of a CNN.

The max-pooling layer reduces the dimension of the output of a CNN and makes the network more robust to frequency variations. The max-pooling operation is applied to the filtered results of CNN only along F dimension to keeping the temporal information along T dimension unaffected.

B Bidirectional Gated Recurrent Unit (GRU) and output layers

Bidirectional GRUs are the bidirectional RNNs which include two hidden layers side-by-side. The input sequence (forward) and its reverse (backward) are considered as input to the first and second units respectively. It provides additional information to the network to learn effectively. Hereafter, this approach is referred to as CBRNN (Convolutional Bi-directional Recurrent Neural Network). The output layer is the fully connected time distributed dense layer, which obtains the output (presence of events) for each time frame. The number of hidden units in the output layer is the same as the number of acoustic event classes in the dataset. A sigmoid activation function is used in the output layer to predict active acoustic events in each time frame. The output of a sigmoid is bounded between [0, 1] and it can be considered as probabilities of an acoustic event being present in a frame.

4.1.3 Evaluation

The performance of the proposed approach is evaluated using real-life dataset : TUT Sound Event 2016 (TUT-SED 2016) development set. This TUT-SED 2016 dataset defines four-fold cross-validation set-up for training and testing. In this work, twenty percent of the training data is used for validation. In this work, a segment-based evaluation metrics: $F1$ score and error rate (ER) are used for polyphonic AED (Mesaros et al., 2016a). An audio segment of one-second is considered for performance evaluation.

The dataset used in this work is developed for DCASE challenge. Each challenge task is associated with the baseline system. Participants of the challenge have to consider respective baseline system as reference for developing their systems. Similarly, the baseline system mentioned in DCASE challenge for polyphonic AED task and most recent approaches as the state-of-the-art are considered to compare the performance of our proposed approach.

- Baseline system: Multiple GMM-based classifiers are trained with MFCC features for polyphonic AED (Mesaros et al., 2016b).

- CNN: 40 monaural log mel band energies are considered as features in the CNN approach (Cakir et al., 2017). The CNN network employed is composed of three CNN layers with 96 filters each, followed by a max-pooling applied over frequency dimension. The output of CNN acts as input to the fully connected layer with number of hidden units equal to the number of acoustic event classes in the acoustic scene.

- RNN: 40 monaural log mel band energies are considered as features in the RNN approach (Cakir et al., 2017). The RNN network employed is composed of three Long Short Term Memory (LSTMs) units each one of which comprises 256 units. The output of the LSTM acts as an input to the fully connected layer with the number of hidden units equal to the number of acoustic event classes in the acoustic scene.

- CRNN: It is a combination of above defined CNN and RNN. The same 40 monaural log mel band energies are considered as features to the CRNN approach (Cakir et al., 2017). The CRNN network employed in this work is composed of three CNN layers with 96 filters each, followed by a max-pooling one applied over frequency dimension. The output of CNN (feature map) was fed as input to the three Long Short Term Memory (LSTMs) units each one comprising 256 units. The output of LSTM acts as an input to the fully connected layer with number of hidden units equal to the number of acoustic event classes in the acoustic scene.

- CBRNN: The CBRNN is an extension of CRNN, that can handle more than one feature types (Adavanne et al., 2017). The 40 binaural log mel band energies are considered as features to the three CNN layers with each one containing 100 filters, followed by max-pooling applied over frequency dimension. The Time Difference Of Arrival (TDOA) features are used as an input to the single CNN layer with 100 filters without max-pooling. Features obtained from CNNs are concatenated and fed as an input to the two bi-directional LSTMs, each of them containing 100 units. The output of LSTM is considered as an input to the fully connected layer with number of hidden units equal to the number of acoustic event classes in the acoustic scene.

Hyperparameters of a proposed CBRNN architecture are selected by executing several experiments over predefined ranges using grid search. We chose hyperparameters of a CBRNN architecture, those leads to the best results on the validation set and the same architecture is used to get the results on the test set.

The 40 mel IFgram features are extracted frame-wise from audio recordings and divided into the sequences of length 61, giving rise to the feature matrices of size 61×40. We also experimented with 20, 60 and 80 mel IFgram features but the best results were obtained with 40 mel IFgram features. 61 frames in a sequence are found to be helpful during experiments. Hyperparameters that give the best results are selected from the set of values given in Table 4.1, using grid search.

Table 4.1: Predefined set of hyperparameters for a CBRNN architecture.

Hyperparameters	Values
# CNN layers	{1, 2, 3, 4}
# Bidirectional GRU layers	{1, 2, 3}
# Hidden units in CNN layer	{32, 64, 128, 256}
# Hidden units in Bidirectional GRU layer	{32, 64, 128, 256}
Size of max-pooling layer	{(1, 2), (1, 3), (1, 5)}
Size of filters in CNN layer	{(3, 3), (5, 5), (7, 7)}
Batch size	{32, 64, 128}
Dropout	{0.10, 0.25, 0.50}

(*,*) represents the operation over Time (T) and Frequency (F) dimensions.

Table 4.2: Performance comparison of polyphonic acoustic event detection using $F1$ score and error rate of the proposed mel IFgram + CBRNN method with other popular methods on the TUT-SED 2016 dataset.

Method	Reference	Error Rate	F1 score
Baseline system	(Mesaros et al., 2016b)	**0.91**	23.7
CNN	(Cakir et al., 2017)	1.09	26.4
RNN	(Cakir et al., 2017)	1.10	29.7
CRNN	(Cakir et al., 2017)	0.93	31.3
CBRNN	(Adavanne et al., 2017)	0.95	35.8
Proposed mel IFgram + CBRNN	Proposed	0.92	**38.7**

The proposed CBRNN framework is decided using the best choice of hyperparameters as follows. The numbers of hidden units in three CNN and two bidirectional GRU are set to 256 and 32 respectively. Each CNN layer is followed by a max-pooling layer along F dimension of size 1×3. The dropout rate is set to 0.25. The size of filters in each CNN layer is set to 3×3. Batch size is set to 32. The loss function is set as binary cross-entropy and Adam is used as the gradient descent optimizer to estimate presence or absence of events in a frame. The proposed network is trained using backpropagation algorithm according to the values of the loss function obtained through processing iterations. During testing, the output of the sigmoid layer are chosen using the fixed threshold value of 0.5 as mentioned in (Cakir et al., 2017). This framework is implemented in Python using the Tensorflow library (Abadi et al., 2016) and features are extracted using the Librosa library (McFee et al., 2015).

The experimental results on TUT-SED 2016 dataset are tabulated in Table 4.2.

The proposed approach had given an average $F1$ score equal to 38.7%, which is an improvement of an absolute 15.0%, 12.3%, 9.0%, 7.4% and 2.9% compared to baseline system, CNN, RNN, CRNN and CBRNN approaches reported in the literature respectively. Furthermore, the average ER of the proposed approach is equal to 0.92, which is considerably lesser than the CNN (-0.17), RNN (-0.18), CRNN (-0.01) and CBRNN (-0.03). However, ER of the baseline system is little lesser than that of the proposed approach (-0.01).

Proposed mel IFgram features + CBRNN approach improves the performance of log mel band energies + CBRNN approach reported in (Adavanne et al., 2017). Our approach uses bidirectional GRUs, which regulate the flow of information with the help of its internal mechanisms known as gates. A number of gates in GRU are less and computationally efficient than bidirectional LSTMs. One can refer to (Dey and Salemt, 2017) for more information about the internal architecture of GRU and LSTM. Proposed mel IFgram features for polyphonic AED incorporate information regarding the rate of change of spectrum rather than just the frequency variation with respect to the time. Mel IFgram features discriminate polyphonic acoustic events more effectively than log mel band energies. Hence the proposed combination of mel IFgram features + CBRNN, outperforms all other approaches with improved $F1$ score.

4.1.4 Contributions and Limitations

Performance of the CRNN is improved using bidirectional GRU and mel IFgram features. Results show that the proposed approach has a significant contribution towards polyphonic AED. However, proposed CBRNN is composed of a considerable number of layers and parameters (more than 1M) and requires a larger dataset for training; if the dataset is not sufficiently large, then this model may be affected by the problem of overfitting. Hence, the proposed approach may not be suitable for smaller datasets.

4.2 A Deep Neural Network-driven feature learning method for polyphonic Acoustic Event Detection from real-life recordings

Recently, DNN-driven feature learning approach was investigated and applied for multi-view Facial Expression Recognition (FER) (Zhang et al., 2016). The authors claim that the DNN-driven feature learning method learns effectively on a smaller dataset and outperforms CNN during FER without overfitting.

In this work, DNN-driven feature learning approach for polyphonic AED is proposed and it is based on the DNN architecture presented in (Zhang et al., 2016). A series of layers including two projection layers, one CNN, two fully connected layers and a sigmoid layer are stacked to construct the proposed DNN model for polyphonic AED. A new projection layer of the proposed DNN model learns the discriminative spectral properties of multiple acoustic events in the mixture with multi channel projection matrices. Further, CNN extracts high-level features, which improves the performance of polyphonic AED.

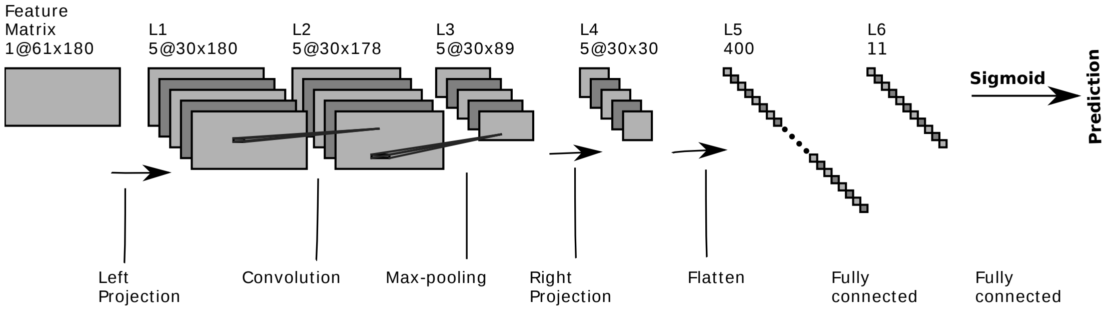


Figure 4.2: Architecture of the proposed DNN-driven feature learning framework for polyphonic Acoustic Event Detection.

4.2.1 Deep Neural Network based framework for polyphonic Acoustic Event Detection

Proposed DNN-driven feature learning model includes two projection layers, one CNN, two fully connected and one sigmoid layers (see Figure 4.2). An F-dimensional feature vectors of log mel band energies (frequency bands) each from

T frames in a sequence forms a feature matrix $X \in \mathbb{R}^{T \times F}$. Feature matrices of the acoustic events are used as input data to train the proposed DNN. Each layer of our DNN is explained below in brief.

A Projection layers

The projection layers are broadly classified into two types as left and right ones. The left projection layer projects the left (temporal, T) dimension of an input feature matrix from one space to another space (see the dimension of the output of the left projection layer in Figure 4.2). This is achieved by taking the dot product between projection matrices, also referred to as left multiplication projection matrices and input feature matrix. The left multiplication projection matrices are integrate the log mel band energies of all sequences (X) of acoustic events to generate more discriminative features for polyphonic AED. Similarly, The right projection layer projects the right (frequency, F) dimension of the feature map (generated by CNN) from one space to another space. Besides, the right multiplication projection matrices are used to extract more significant features further from higher-level feature maps. Projection layers work in the same way as spatial filtering in computer vision (Griffith, 2013) and highlight the energy variations of the acoustic events before and after CNN. The weights of the projection matrices are initialized randomly using uniform distribution as done in (Cakir et al., 2016). Let $\mathcal{H}_k = \{H_{k,i}^{(l)} | i = 1, ..., M_l\}$ $(k = 1, ..., N_l)$ represent the kth set of M_l channels (multi-channel) of the projection matrices. Where, $H_{k,i}^{(l)}$ is the ith channel projection matrix of $\mathcal{H}_k$, M_l is the number of channels in $\mathcal{H}_k$ and N_l is the number of sets of multi-channel projection matrices. Further, left projection layer can be formulated as

$$O_k = \sum_{i=1}^{M_l} H_{k,i}^{(l)} X_i, \ (k = 1, ..., N_l) \tag{4.6}$$

Where O_k is the matrix in the kth output channel, X_i is the input matrices of the ith channel. Likewise, right multiplication projection layer can be formulated as

$$O_k = \sum_{i=1}^{M_r} X_i H_{k,i}^{(r)}, \ (k = 1, ..., N_r) \tag{4.7}$$

Projection layers project input feature matrix X from one space to the another and reduce the dimension of the outcome simultaneously.

B CNN layer

The output of the left multiplication projection layer is fed as an input to the CNN layer. CNN layer processes a small local regions of the input using the group of filters followed by the max-pooling layer. Unlike well-known CNN approaches used for polyphonic AED, the filters used in this work are one-dimensional sequences, convolved only along F dimension of the input feature matrix. These filters may get a stronger response from some local regions of the input feature matrix, while other regions are suppressed. This phenomenon captures the high energy spectral components of the acoustic events from the input feature matrix.

CNN outputs the higher level feature maps from the matrices of left multiplication projection layer. It is to be noted here that, left and right multiplication projection layers reduce the number of connections (layers) compared to the classic two-dimensional CNN to avoid overfitting problem (Mobahi et al., 2009).

The feature maps from CNN pass through tanh activation function before fed as input to the max-pooling layer, which reduces the dimension and makes the network robust to frequency variations. The max-pooling operation is applied to the filtered results of CNN only along F dimension.

The output of the right multiplication projection layer is represented as $P = [P_{c,t,f}]C \times T \times F'$, where C, T, F' are the number of channels, rows and columns respectively, available after right multiplication projection.

C Fully connected and sigmoid layers

The fully connected and sigmoid layers are used as they were used in classical CNN. The inputs from the previous layer are combined with the fully connected layer and sigmoid layer predicts several active acoustic events simultaneously. The binary cross-entropy is used as a loss function (Bulat and Tzimiropoulos, 2016). The proposed network is trained using backpropagation algorithm based on the

Table 4.3: Performance comparison of polyphonic acoustic event detection using the proposed DNN framework with other methods on the TUT-SED 2016 dataset.

Method	Reference	Error Rate	F1 score
Baseline system	(Mesaros et al., 2016b)	0.91	23.7
CNN	(Cakir et al., 2017)	1.09	26.4
RNN	(Cakir et al., 2017)	1.10	29.7
CRNN	(Cakir et al., 2017)	0.93	31.3
CBRNN	(Adavanne et al., 2017)	0.95	35.8
mel IFgram + CBRNN	Mulimani & Koolagudi: previous approach	0.92	38.7
Proposed DNN	Proposed	**0.71**	**44.7**

values of the loss function obtained through successive iterations. During testing, outputs of the sigmoid layer are chosen using the fixed threshold value of 0.5.

4.2.2 Evaluation

The parameters which give the best results on the validation set are selected using the grid search as explained in our previous CBRNN approach. The proposed DNN framework with its best parameters is organized as follows. 60 monaural (one channel) log mel band energies, their deltas and acceleration features are extracted frame-by-frame from audio recordings and divided into the sequences of length 61, giving rise to the feature matrices of size 61×180. The left projection layer contains projection matrices of size $5 \times 30 \times 61 \times 1$, indicating that there are 5 single channel matrices with 30 rows and 61 columns. Size of the filter in CNN layer is $5 \times 1 \times 3 \times 5$, which has the number of channels equal to the number of single channel matrices in the left projection layer. The right projection layer contains projection matrices of size $5 \times 89 \times 30 \times 5$. The output of the right projection layer is converted into a long vector and is fed as an input to the first fully connected layer, which has the transformation matrix of size 4500×40 (40 is the batch size). This transforms the dimension of the feature vectors from 4500 to 40. The second fully connected layer has the transformation matrix of size $40 \times C$, where C denotes the number of classes in the acoustic scene. This DNN framework is implemented in Python using the Tensorflow library (Abadi et al., 2016) and features are extracted using the Librosa library (McFee et al., 2015).

The experimental results on TUT-SED 2016 dataset are tabulated in Table 4.3. Proposed DNN framework achieves an average $F1$ score equal to 44.7%, which is an improvement of an absolute 20.8%, 18.1%, 14.8%, 13.2%, 8.7% and 5.8% compared to the baseline system, CNN, RNN, CRNN, CBRNN and improved CBRNN (mel IFgram + CBRNN) approaches respectively. Furthermore, the average ER of the proposed DNN framework is equal to 0.71, which is considerably lesser than the baseline system (-0.2), CNN (-0.38), RNN (-0.39), CRNN (-0.22), CBRNN (-0.24) and improved CBRNN (-0.21). The TUT-SED 2016 dataset contains a small amount of real-time audio recordings and acoustic events occur sparsely (a maximum portion of the audio recordings is silent). Normally, the CRNN, CBRNN and improved CBRNN are said to learn effectively with a larger dataset and present dataset may not be sufficient for their training. This may be the reason for poor performance of those approaches compared to the proposed method. However, the combination of projection layers and CNN layer extracts more discriminative spectral features from multiple acoustic events in a mixture and exhibits improved performance over the traditional CNN and CBRNN. Better results also convey the suitability of the proposed DNN for smaller datasets.

4.2.3 Contributions and Limitations

The main contributions of this work are summarized below.

- A new projection layer is introduced with the CNN for polyphonic AED and it is helpful to learn the discriminative spectral characteristics of polyphonic acoustic events. Notably, significant features per frame associated with multiple overlapped acoustic events in the mixture are integrated to produce ensemble set of more discriminative features.

- Unlike two-dimensional CNN layers used for polyphonic AED, in this work, the one-dimensional CNN layer is employed to extract high-level features from the input. It is empirically observed that the proposed projection and CNN layers for polyphonic AED significantly reduce the complexity of the DNN model and alleviate the overfitting.

A main limitation of the proposed DNN model is that CNNs fail to model the long term temporal information prevailing in the acoustic events; this is problematic especially during modeling of long events, such as rain, baby crying, crowd cheering and so on.

4.3 Summary

In this chapter, an attempt to detect the overlapped (polyphonic) acoustic events at a given time instance was undertaken. There were two deep learning models presented for polyphonic AED. One was CRNN and the other was a DNN-driven feature learning approach. Proposed CRNN was a combination of CNN and bidirectional GRU, also referred to as CBRNN, which captured both spectral and temporal information from the polyphonic acoustic events. Further, mel IFgram features were explored and used as input to the CBRNN. It is seen that the combination of mel IFgram features and CBRNN outperforms other popular state-of-the-art approaches.

Two projection layers before (left) and after (right) 1-D CNN were explored in a DNN-driven feature learning approach. A combination of projection layers and CNN extracted more discriminative features from polyphonic acoustic events than traditional CNN alone. It is seen that the proposed approach outperforms the CBRNN approach proposed earlier. The advantage of the DNN-driven feature learning approach over CBRNN is that the DNN-driven feature learning approach achieves better performance with lesser number of layers and parameters. The disadvantage is that the DNN-driven feature learning approach fails to capture temporal information.

CHAPTER 5

Acoustic Scene Classification

In chapter 4, two deep learning models: CBRNN and DNN-driven feature learning methods were presented for polyphonic AED. Features extracted from the single channel of an audio signal are used as input features to the deep learning models. In this chapter, the task is to recognize polyphonic acoustic events and the corresponding scene from a continuous audio signal. This is also referred to as joint polyphonic AED and ASC. The CBRNN proposed in a previous chapter is used for joint polyphonic AED and ASC. CBRNN captures both spectral and temporal information from an audio signal. It requires a larger dataset and dataset used in the present studies is sufficient enough to train CBRNN.

Polyphonic acoustic events and respective scenes may be recognized better with features from multi-channels. Hence, in this chapter binaural features (features from two channels) are also explored. An attempt is made to further improve the performance of CBRNN by replacing CNN in CBRNN by Kervolutional Neural Network (KNN) and resulting architecture is referred to as Kervolutional Bidirectional Recurrent Neural Network (KBRNN).

5.1 Design of Convolutional Bidirectional Recurrent Neural Network for joint polyphonic Acoustic Event Detection and Acoustic Scene Classification

The same architecture of CBRNN, proposed in our previous chapter (see Figure 4.1) is used for joint polyphonic AED and ASC. Target label matrix $Y \in \mathbb{R}^{(C+\hat{C})\times T}$ of the CBRNN network for joint polyphonic AED and ASC is a combination of both acoustic events C and corresponding scenes $\hat{C}$, in contrast to the label matrix used in polyphonic AED approach, which represents only acoustic events present in a time frame, t. Input features to the CBRNN are broadly categorized into two types. One is monaural features, which are extracted from a single channel and another is binaural features, which are extracted from binaural channels of an audio signal. In this work, Log mel band energies and mel IFgram features are extracted as monaural and binaural features to the CBRNN approach for the joint polyphonic AED and ASC.

5.1.1 Evaluation

The performance of the proposed approach is evaluated using joint sound event and scene dataset. This dataset defines a five-fold cross-validation set-up for training and testing. In this work, a quarter of the training data is used for validation during training. A segment-based evaluation metrics: $F1$ score and ER are used for polyphonic AED (Mesaros et al., 2016a). Majority voting based accuracy is used as a metric for ASC. Majority voting represents the acoustic scene identified in the majority of the frames of an audio signal. A segment of one-second is considered for performance evaluation. During testing, the outputs of the sigmoid layer are chosen using the fixed threshold value of 0.9 as in (Bear et al., 2019). Performance of our proposed approach is compared with the state-of-the-art CRNN approach (Bear et al., 2019) and its architecture is detailed in the previous chapter. 128 mel band energies are considered as input features to the CRNN as reported in the literature for joint polyphonic AED and AEC. In this work, 40 log mel band energies and mel IFgram features are extracted and used as input features to the CBRNN. We also had experimented with 20, 60 and 80 features but the best results were obtained with 40 features. The experimental results of the proposed approach are tabulated in Table 5.1. A combination of monaural/binaural features + CBRNN outperforms state-of-the-art CRNN. Binaural features are a combination (concatenation) of significant information from both the channels of the audio signals which improves the performance of CBRNN over that of the monaural features.

Table 5.1: Performance comparison of joint polyphonic AED and ASC using a combination of monaural and binaural features + CBRNN with state-of-the-art CRNN method.

			Polyphonic AED		ASC
Type of features	Method	Reference	Error Rate	F1 score	Acc.
Monaural	Log mel band energies + CRNN	(Bear et al., 2019)	1.00	13.8	98.0
	Log mel band energies + CBRNN	-	0.98	14.2	98.0
	mel IFgram + CBRNN	-	0.83	18.9	98.3
Binaural	Log mel band energies and log mel band energies + CBRNN	-	0.97	15.1	98.0
	mel IFgram and mel IFgram + CBRNN	-	0.81	19.2	98.3
	mel IFgram and log mel band energies + CBRNN	-	**0.78**	**21.3**	**98.5**

One can observe from Table 5.1 that, different combinations of binaural features are considered to improve the performance of both polyphonic AED and ASC. As mentioned in our previous chapter, mel IFgram features perform better over log mel band energies. Monaural mel IFgram features outperform both monaural and binaural log mel band energies. Binaural mel IFgram features improve $F1$ score and ER of a polyphonic AED task compared to monaural mel IFgram features. However, the recognition accuracy (98%) of the ASC task is unchanged.

Further, we concatenate mel IFgram features from the first channel and log mel band energies from the second channel of an audio signal. These binaural features combine both magnitude (log mel band energies) and phase (mel IFgram) information. Combining mel IFgram features with log mel band energies provides an absolute improvement in $F1$ score by 2.4%, lesser ER (-0.05) and improved accuracy by 0.2% as compared to the performance of mel IFgram features alone. Hence, the combination of mel IFgram features with log mel band energies suits well for joint polyphonic AED and ASC.

Bidirectional GRU provides additional significant information to the network using forward and backward sequences as compared to unidirectional LSTM. Hence, the proposed combination of mel IFgram features and log mel band energies with CBRNN detects acoustic events with an average $F1$ score equal to 21.3%, which is a significant improvement of an absolute 7.5% over the state-of-the-art CRNN approach reported in the literature. Furthermore, the average ER of the proposed approach is equal to 0.78, which is considerably lesser than the CRNN (-0.22). The proposed approach also recognizes the acoustic events with a recognition accuracy of 98.5% which is slightly better than that of the CRNN (0.5%).

5.1.2 Contributions and Limitations

In this work, binaural features are explored for joint polyphonic AED and ASC. Binaural mel IFgram features and log mel band energies improve the performance of CBRNN as compared to monaural and other binaural features. However, CNN layers in CBRNN are linear and their convolutional operations may be generalized to non-linear operations for better performance.

5.2 Design of Kervolutional Bidirectional Recurrent Neural Network for joint polyphonic Acoustic Event Detection and Acoustic Scene Classification

Convolutional layers are linear and non-linearity is added to them by activation functions, such as ReLU. However, activation functions provide point-wise non-linearity only. Performance of the CNN may be further improved by generalizing the convolutional operation to patch-wise (region-wise) non-linear operation using kernel trick. This operation is also known as kervolutional (**ker**nel con**volutional**) (Wang et al., 2019).

5.2.1 Architecture of proposed Kervolutional Recurrent Neural Network

The architecture of the proposed Kervolutional Bidirectional Recurrent Neural Network (KBRNN) is the same as that of the architecture of CBRNN proposed in our previous chapter (shown in Figure 4.1) except KBRNN uses Kervolutional Neural Network (KNN) than simple CNN.

A KNN layer

Convolutional operation is denoted using the notation given in (5.1).

$$Z = X \bigoplus f \tag{5.1}$$

Where $X \in \mathbb{R}^{F \times T}$ is a feature matrix, f is a filter and $\bigoplus$ is the convolutional operation. The output of an i^{th} element of a convolutional operation is denoted below in (5.2).

$$z_i = (x_i, f) \tag{5.2}$$

Where $(*, *)$ denotes the inner product between two vectors. In the similar way, Kervolutional operation is denoted using notation given in (5.3).

$$Z = X \bigotimes f \tag{5.3}$$

Where $\bigotimes$ is the kervolutional operation. The output of an i^{th} element of a kervolutional operation is denoted as given in (5.4).

$$z_i = (\psi(x_i), \psi(f)) \tag{5.4}$$

Where ψ is a non-linear function and it is computed using kernel trick given in (5.5).

$$(\psi(x_i), \psi(f)) = \sum_j c_j (x_i^T f)^j = \kappa(x_i, f) \tag{5.5}$$

Where c_j is the coefficient, which balances the order of non-linearity and κ is the kernel function. In this work, the linearity of CNN is replaced with the polynomial kernel and its formulation is given in (5.6).

$$\kappa_p(x, f) = (x^T f + c_p)^{d_p} = \sum_{j=0}^{d_p} c_p^{d_p - j}(x^T f)^j \tag{5.6}$$

Where d_p denotes the order of polynomial and it converts the dimension of features from one space to the other. We only replace the CNN layers of CBRNN by the kervolutional layer using the polynomial kernel and the resulting architecture is referred to as KBRNN.

5.2.2 Evaluation

Values of parameters, d_p and c_p of polynomial kernel that give the best results are selected from the sets $d_p = \{2, 3, 5, 7\}$ and $c_p = \{0.5, 1\}$ using grid search. Best results were obtained with $d_p = 3$ and $c_p = 0.5$. Binaural mel IFgram features and log mel band energies have shown to be significant and suitable features for joint polyphonic AED and ASC in our previous approach. Hence, they are again considered as input features to the proposed KBRNN approach.

Table 5.2: Performance comparison of joint polyphonic AED and ASC using binaural mel IFgram and log mel band energies with KBRNN and CBRNN approach.

		Polyphonic AED		ASC
Method	Reference	Error Rate	F1 score	Acc.
mel IFgram and log mel band energies + CBRNN	Mulimani & Koolagudi: previous approach	0.78	21.3	98.5
mel IFgram and log mel band energies + KBRNN	Proposed	**0.75**	**24.2**	**99.0**

The experimental results of the proposed approach are tabulated in Table 5.2. The replacement of CNN by KNN significantly improves the performance of CBRNN for joint polyphonic AED and ASC. The proposed KBRNN approach achieves an average $F1$ score equal to 24.2%, which is an absolute improvement of 2.9% compared to our CBRNN approach. Furthermore, the average ER of the proposed

KBRNN approach is equal to 0.75, which is lesser than the CBRNN (-0.03). The proposed KBRNN also recognizes the corresponding acoustic scenes with an average accuracy of 99%, which is slightly higher than the CBRNN (0.5%). KNN using polynomial kernel captures higher-order interactions of the binaural features and performs better than CNN. Hence, the proposed KBRNN is more suitable for joint polyphonic AED and ASC.

5.2.3 Contributions and Limitations

In this work, KNN is explored for joint polyphonic AED and ASC. The traditional linear convolutional operation is generalized using the non-linear polynomial kernel. Binaural features + combination of KNN and bidirectional GRU (KBRNN) have significant contribution towards joint polyphonic AED and ASC. However, in this work, only one non-linear polynomial kernel is used for the computation of KNN. Other non-linear functions may be explored for further investigation.

5.3 Summary

In this chapter, an attempt for the recognition of polyphonic acoustic events and a related scene from a continuous audio signal is undertaken. Different combinations of binaural features are explored and considered as input features to the CBRNN for joint polyphonic AED and ASC. A combination of mel IFgram features and log mel band energies improves the performance of CBRNN as compared to other binaural and monaural features.

Linear convolutional operations of CNN layers in CBRNN are generalized to non-linear operations using the polynomial kernel and the resulting layers are known as KNN layers. CNN in CBRNN is replaced by KNN and resulting KBRNN is used for joint polyphonic AED and ASC. The polynomial kernel in KNN performs better than traditional CNN. A combination of KNN and bidirectional GRU, outperforms CBRNN.

CHAPTER 6

Summary and Conclusions

The thesis is organized into 6 chapters. The first chapter introduces monophonic Acoustic Event Classification (AEC), polyphonic Acoustic Event Detection (AED) and joint model (polyphonic AED and ASC) with their applications and challenges in brief. The second chapter critically reviews the research work done in the area of monophonic AEC, polyphonic AED and joint model concerning different features and classifiers. At the end of this chapter, research gaps are analyzed and problem statement is identified. In the third chapter, Spectrogram Features (SFs), Spectrogram Image Features (SIFs) and MapReduce-based Features (MRFs) are proposed for monophonic AEC. The chapter four, presents two deep learning models: one is Convolutional Bidirectional Recurrent Neural Network (CBRNN) and the other is the DNN-driven feature learning approach for polyphonic AED. Chapter five explores binaural features and Kervolutional Neural Network (KNN) to improve the performance of CBRNN for joint polyphonic AED and ASC. Chapter six concludes the present work and opens up the path for further research.

6.1 Summary of the Present Work

In this thesis, effective methods for monophonic AEC, polyphonic AED and joint model (polyphonic AED and ASC) are investigated. Frame-based speech features are specifically designed and developed for speech/speaker recognition tasks. This research proposes acoustic event specific features such as SFs, SIFs and MRFs for monophonic AEC. SIFs such as Bag-of-Visual-Words (BoVWs), Fusion Fisher Vector (FFV) features and Fusion-based Bag-of-Features (FBoFs) are robust to noise and outperform both baseline and state-of-the-art methods. Results show that FBoFs achieve a relatively higher performance than BoVWs and FFV features. MRFs are specially designed and developed from more massive audio datasets. Even MRFs outperform state-of-the-art methods.

A combination of mel IFgram features and CBRNN is proposed for polyphonic AED and it outperforms the popular state-of-the-art approaches. CBRNN requires a larger dataset for effective training. The present TUT-SED 2016 development dataset may not be sufficient to train CBRNN. In this thesis, a DNN-driven feature learning method is proposed for polyphonic AED, which is well suitable for smaller datasets and outperforms state-of-the-art methods. CBRNN is also used for joint polyphonic AED and ASC. The dataset used for this task is sufficiently large enough to train CBRNN effectively. Further, the performance CBRNN is improved by computing and using binaural features as input. Binaural features outperform the monaural features. CNN in CBRNN is replaced by KNN to get Kervolutional Bidirectional Recurrent Neural Network (KBRNN). The proposed KBRNN for joint polyphonic AED and ASC outperforms the CBRNN.

6.2 Conclusions

- The strongest peaks of acoustic event signals are unaffected by noise and their properties are effectively discriminated by BoVWs, FFV features, FBoFs and MRFs. Hence, the proposed SIFs and MRFs are robust in nature.

- GTCCs and their BoAW representations outperform MFCCs. However, BoAW representations of GTCCs do not perform alone as expected. A fusion of BoAW representations of GTCCs and the proposed BoVWs generates more robust features for monophonic AEC, in noisy conditions as compared to BoVWs alone and FFV features.

- MapReduce job for feature extraction, which has been especially designed for more massive audio datasets and may not be suitable for smaller datasets.

- Chi-square kernel of SVM classifier is computed from nature of the input features, hence, Chi-square SVM effectively classifies the histogram (bag) features than linear SVM. However, the computation of Chi-square demands higher computational time as compared to the linear kernel. FFV features perform well with a linear classifier and include much more information as compared to BoVWs.

- The proposed SFs, SIFs and MRFs have shown considerably good recognition accuracy in classifying monophonic AEC. It indicates that proposed features have a significant contribution towards the AEC.

- CBRNN is a combination of CNN and bidirectional GRU. Bidirectional GRU learns from both the input sequence and its reverse. Hence, CBRNN learns from additional information and perform better than state-of-the-art CRNN.

- It is observed that projection layers of proposed DNN framework have more discriminative ability while detecting overlapped acoustic events.

- A DNN-driven feature learning method effectively recognizes the overlapped acoustic events from smaller datasets but fails to capture temporal information from the acoustic signals. Acoustic events with long term temporal context such as baby crying, rain sound may not be recognized effectively.

- Polyphonic acoustic events and corresponding scenes are effectively recognized with binaural features. Binaural features are computed by concatenating the significant information from both the channels. Hence, binaural features outperform monaural features.

- Replacement of CNN in CBRNN by KNN effectively recognizes polyphonic acoustic events and corresponding scenes. Non-linear operations, using a polynomial kernel, are more effective than linear convolutional operations.

6.3 Scope for Future Works

- In the current research, a left singular vector is divided into two parts using an identified threshold. One part represents spectral components belonging to an event and the other represents the spectral components belonging to a non-event. In future, a more effective threshold may be defined, that divides singular vector into two parts. One part represents spectral components belonging to an event and the other represents the spectral components belonging to noise and non-event. This extension may generate noise robust SFs.
- Current BoVWs are computed using hard quantization. A GMMs may be used for soft quantization in place of K-means clustering; this may further improve the recognition accuracy.
- BoVWs representation is an unordered histogram representation, which does not include temporal information. Temporal information is an important cue for acoustic event characterization. Loss of temporal information can be avoided using temporal pyramid or feature augmentation (Grzeszick et al., 2017).
- In the future, the Fisher vector may be computed from another Fisher vector of a monochrome image hierarchically for more discriminative FFV features for robust AEC. The computation of the Fisher vector from another Fisher vector may be referred to as deep Fisher network.
- Combination of the Fisher vectors from monochrome images of different colormaps may improve the performance of the AEC system.
- The advanced dimension reduction approaches, such as Linear Discriminant Analysis (LDA) and max-margin learning, may be explored in the place of traditional PCA for selection of significant features from Fisher vectors.
- Proposed DNN for polyphonic AED fails to model long term temporal information. In the future, RNN may be added to the proposed DNN framework

in order to consider temporal information.

- CBRNN/KBRNN learns effectively on larger datasets. In the future, Generative Adversarial Networks (GAN) and Variational Auto Encoders (VAE) may be considered to create additional training examples (data augmentation) in smaller datasets, such as, TUT-SED 2016 development set.
- Different combinations of binaural features and different non-linear kernel tricks may improve the overall performance of KBRNN for joint polyphonic AED and ASC.

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283.

Abe, T., Kobayashi, T., and Imai, S. (1995). Harmonics tracking and pitch extraction based on instantaneous frequency. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 756–759. IEEE.

Adavanne, S., Pertilä, P., and Virtanen, T. (2017). Sound event detection using spatial features and convolutional recurrent neural network. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 333–336. IEEE.

Almeer, M. H. (2012). Cloud Hadoop MapReduce for remote sensing image analysis. *Journal of Emerging Trends in Computing and Information Sciences*, 3(4):637–644.

Bae, S. H., Choi, I., and Kim, N. S. (2016). Acoustic scene classification using parallel combination of lstm and cnn. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, pages 11–15.

Barchiesi, D., Giannoulis, D., Stowell, D., and Plumbley, M. D. (2015). Acoustic scene classification: Classifying environments from the sounds they produce. *IEEE Signal Processing Magazine*, 32(3):16–34.

Bear, H. L., Nolasco, I., and Benetos, E. (2019). Towards joint sound scene and polyphonic sound event recognition. *arXiv preprint arXiv:1904.10408*.

Bhuiyan, M. A. and Al Hasan, M. (2015). An iterative MapReduce based frequent subgraph mining algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 27(3):608–620.

Boashash, B., Azemi, G., and Khan, N. A. (2015). Principles of time–frequency feature extraction for change detection in non-stationary signals: Applications to newborn EEG abnormality detection. *Pattern Recognition*, 48(3):616–627.

Bulat, A. and Tzimiropoulos, G. (2016). Human pose estimation via convolutional part heatmap regression. In *European Conference on Computer Vision*, pages 717–732. Springer.

Cakir, E., Heittola, T., Huttunen, H., and Virtanen, T. (2015). Polyphonic sound event detection using multi label deep neural networks. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

Cakir, E., Ozan, E. C., and Virtanen, T. (2016). Filterbank learning for deep neural network based polyphonic sound event detection. In *International Joint Conference on Neural Networks (IJCNN)*, pages 3399–3406. IEEE.

Cakir, E., Parascandolo, G., Heittola, T., Huttunen, H., Virtanen, T., Cakir, E., Parascandolo, G., Heittola, T., Huttunen, H., and Virtanen, T. (2017). Convolutional recurrent neural networks for polyphonic sound event detection. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 25(6):1291–1303.

Carletti, V., Foggia, P., Percannella, G., Saggese, A., Strisciuglio, N., and Vento, M. (2013). Audio surveillance using a bag of aural words classifier. In *10th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 81–86. IEEE.

Cauchi, B., Siedenburg, K., Santos, J. F., Falk, T. H., Doclo, S., and Goetze, S. (2019). Non-intrusive speech quality prediction using modulation energies and lstm-network. *IEEE/ACM Transactions on Audio, Speech, and Language Processing.*

Cernekova, Z., Kotropoulos, C., and Pitas, I. (2003). Video shot segmentation using singular value decomposition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages 176–181.

Chen, B.-W., Chen, C.-Y., and Wang, J.-F. (2013). Smart homecare surveillance system: Behavior identification based on state-transition support vector machines and sound directivity pattern analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(6):1279–1289.

Cheng, O., Abdulla, W., and Salcic, Z. (2005). Performance evaluation of front-end processing for speech recognition systems. *The University of Auckland, Report 621*.

Choi, K., Fazekas, G., Sandler, M., and Cho, K. (2017). Convolutional recurrent neural networks for music classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2392–2396. IEEE.

Chu, S., Narayanan, S., and Kuo, C. C. J. (2009). Environmental sound recognition with time–frequency audio features. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(6):1142–1158.

Conte, D., Foggia, P., Percannella, G., Saggese, A., and Vento, M. (2012). An ensemble of rejecting classifiers for anomaly detection of audio events. In *IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 76–81. IEEE.

Cooke, M. (2006). A glimpsing model of speech perception in noise. *The Journal of the Acoustical Society of America*, 119(3):1562–1573.

Cooke, M., Green, P., Josifovski, L., and Vizinho, A. (2001). Robust automatic speech recognition with missing and unreliable acoustic data. *Speech communication*, 34(3):267–285.

Cotton, C. V. and Ellis, D. P. (2011). Spectral vs. spectro-temporal features for acoustic event detection. In *2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 69–72. IEEE.

Cristani, M., Bicego, M., and Murino, V. (2007). Audio-visual event recognition in surveillance video sequences. *IEEE Transactions on Multimedia*, 9(2):257–267.

Crocco, M., Cristani, M., Trucco, A., and Murino, V. (2016). Audio surveillance: a systematic review. *ACM Computing Surveys (CSUR)*, 48(4):52.

Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal processing*, 28(4):357–366.

Dean, J. and Ghemawat, S. (2008). Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.

Deng, Z., Peng, X., Li, Z., and Qiao, Y. (2019). Mutual component convolutional neural networks for heterogeneous face recognition. *IEEE Transactions on Image Processing*.

Dennis, J., Tran, H. D., and Chng, E. S. (2013a). Image feature representation of the subband power distribution for robust sound event classification. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(2):367–377.

Dennis, J., Tran, H. D., and Chng, E. S. (2013b). Overlapping sound event recognition using local spectrogram features and the generalised hough transform. *Pattern Recognition Letters*, 34(9):1085–1093.

Dennis, J., Tran, H. D., and Li, H. (2011). Spectrogram image feature for sound event classification in mismatched conditions. *IEEE Signal Processing Letters*, 18(2):130–133.

Dey, R. and Salemt, F. M. (2017). Gate-variants of gated recurrent unit (gru) neural networks. In *International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1597–1600. IEEE.

Eronen, A. J., Peltonen, V. T., Tuomi, J. T., Klapuri, A. P., Fagerlund, S., Sorsa, T., Lorho, G., and Huopaniemi, J. (2006). Audio-based context recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):321–329.

Foggia, P., Petkov, N., Saggese, A., Strisciuglio, N., and Vento, M. (2015). Reliable detection of audio events in highly noisy environments. *Pattern Recognition Letters*, 65:22–28.

Foggia, P., Petkov, N., Saggese, A., Strisciuglio, N., and Vento, M. (2016). Audio surveillance of roads: a system for detecting anomalous sounds. *IEEE Transactions on Intelligent Transportation Systems*, 17(1):279–288.

Foster, P., Sigtia, S., Krstulovic, S., Barker, J., and Plumbley, M. D. (2015). Chime-home: A dataset for sound source recognition in a domestic environment. In *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 1–5. IEEE.

Freitag, M., Amiriparian, S., Pugachevskiy, S., Cummins, N., and Schuller, B. (2017). audeep: Unsupervised learning of representations from audio with deep recurrent neural networks. *The Journal of Machine Learning Research*, 18(1):6340–6344.

Geiger, J. T., Schuller, B., and Rigoll, G. (2013). Large-scale audio feature extraction and svm for acoustic scene classification. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 1–4. IEEE.

Gerhard, D. (2003). *Audio signal classification: History and current techniques*. Citeseer.

Ghoraani, B. and Krishnan, S. (2011). Time–frequency matrix feature extraction and classification of environmental audio signals. *IEEE transactions on audio, speech, and language processing*, 19(7):2197–2209.

Goetze, S., Schroder, J., Gerlach, S., Hollosi, D., Appell, J.-E., and Wallhoff, F. (2012). Acoustic monitoring and localization for social care. *Journal of Computing Science and Engineering*, 6(1):40–50.

Golub, G. H. and Reinsch, C. (1970). Singular value decomposition and least squares solutions. *Numerische mathematik*, 14(5):403–420.

Gorin, A., Makhazhanov, N., and Shmyrev, N. (2016). Dcase 2016 sound event detection system based on convolutional neural network. In *IEEE AASP Challenge: Detection and Classification of Acoustic Scenes and Events.*

Griffith, D. A. (2013). *Spatial autocorrelation and spatial filtering: gaining understanding through theory and scientific visualization.* Springer Science & Business Media.

Grzeszick, R., Plinge, A., and Fink, G. A. (2017). Bag-of-features methods for acoustic event detection and classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(6):1242–1252.

Guo, G. and Li, S. Z. (2003). Content-based audio classification and retrieval by support vector machines. *IEEE Transactions on Neural Networks*, 14(1):209–215.

Harma, A., McKinney, M. F., and Skowronek, J. (2005). Automatic surveillance of the acoustic activity in our living environment. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 4–8. IEEE.

Heikkinen, A., Sarvanko, J., Rautiainen, M., and Ylianttila, M. (2013). Distributed multimedia content analysis with mapreduce. In *2013 IEEE 24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, pages 3497–3501. IEEE.

Heil, C. E. and Walnut, D. F. (1989). Continuous and discrete wavelet transforms. *SIAM review*, 31(4):628–666.

Heittola, T., Mesaros, A., Eronen, A., and Virtanen, T. (2010). Audio context recognition using audio event histograms. In *18th European Signal Processing Conference*, pages 1272–1276. IEEE.

Heittola, T., Mesaros, A., Virtanen, T., and Eronen, A. (2011). Sound event detection in multisource environments using source separation. In *CHIME 2011 Workshop on Machine Listening in Multisource Environments*, pages 36–40.

Hlawatsch, F. and Boudreaux-Bartels, G. F. (1992). Linear and quadratic time-frequency signal representations. *IEEE signal processing magazine*, 9(2):21–67.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456.

Jaakkola, T. and Haussler, D. (1999). Exploiting generative models in discriminative classifiers. In *Advances in neural information processing systems*, pages 487–493.

Jayalakshmi, S., Chandrakala, S., and Nedunchelian, R. (2018). Global statistical features-based approach for acoustic event detection. *Applied Acoustics*, 139:113–118.

Jolliffe, I. T. (1986). Choosing a subset of principal components or variables. In *Principal Component Analysis*, pages 92–114. Springer.

Karlik, B. and Olgac, A. V. (2011). Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 1(4):111–122.

Kiktova-Vozarikova, E., Juhar, J., and Cizmar, A. (2015). Feature selection for acoustic events detection. *Multimedia Tools and Applications*, 74(12):4213–4233.

Kim, K. and Ko, H. (2011). Hierarchical approach for abnormal acoustic event classification in an elevator. In *Proceedings of the 8th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 89–94. IEEE.

Kim, Y. and Mesbahi, M. (2006). On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian. *IEEE transactions on Automatic Control*, 51(1):116–120.

Klema, V. and Laub, A. (1980). The singular value decomposition: Its computation and some applications. *IEEE transactions on automatic control*, 25(2):164–176.

Kong, Q., Sobieraj, I., Wang, W., and Plumbley, M. (2016). Deep neural network baseline for DCASE challenge 2016. *Proceedings of DCASE*.

Kons, Z., Toledo-Ronen, O., and Carmel, M. (2013). Audio event classification using deep neural networks. In *Interspeech*, pages 1482–1486.

Kürby, J., Grzeszick, R., Plinge, A., and Fink, G. A. (2016). Bag-of-features acoustic event detection for sensor networks. In *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, pages 55–59.

Li, J., Dai, W., Metze, F., Qu, S., and Das, S. (2017). A comparison of deep learning methods for environmental sound detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 126–130. IEEE.

Lojka, M., Pleva, M., Kiktová, E., Juhár, J., and Čižmár, A. (2016). Efficient acoustic detector of gunshots and glass breaking. *Multimedia Tools and Applications*, 75(17):10441–10469.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

Ludeña-Choez, J. and Gallardo-Antolín, A. (2016). Acoustic event classification using spectral band selection and non-negative matrix factorization-based features. *Expert Systems with Applications*, 46:77–86.

Makhoul, J. (1975). Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580.

Malkin, R. G. and Waibel, A. (2005). Classifying user environment for mobile applications using linear autoencoding of ambient audio. In *IEEE International*

Conference on Acoustics, Speech, and Signal Processing (ICASSP), volume 5, pages v–509. IEEE.

Manoochehri, M. (2013). *Data just right: introduction to large-scale data & analytics*. Addison-Wesley.

Maxime, J., Alameda-Pineda, X., Girin, L., and Horaud, R. (2014). Sound representation and classification benchmark for domestic robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6285–6292. IEEE.

McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., and Nieto, O. (2015). librosa: Audio and music signal analysis in python. In *Proceedings of SciPy*, pages 18–25.

McLoughlin, I., Zhang, H., Xie, Z., Song, Y., and Xiao, W. (2015). Robust sound event classification using deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):540–552.

Merris, R. (1994). Laplacian matrices of graphs: a survey. *Linear algebra and its applications*, 197:143–176.

Mesaros, A., Heittola, T., Dikmen, O., and Virtanen, T. (2015). Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 151–155. IEEE.

Mesaros, A., Heittola, T., Diment, A., Elizalde, B., Shah, A., Vincent, E., Raj, B., and Virtanen, T. (2017). Dcase 2017 challenge setup: Tasks, datasets and baseline system. In *DCASE 2017-Workshop on Detection and Classification of Acoustic Scenes and Events*.

Mesaros, A., Heittola, T., Eronen, A., and Virtanen, T. (2010). Acoustic event detection in real life recordings. In *18th European Signal Processing Conference*, pages 1267–1271. IEEE.

Mesaros, A., Heittola, T., and Virtanen, T. (2016a). Metrics for polyphonic sound event detection. *Applied Sciences*, 6(6):162.

Mesaros, A., Heittola, T., and Virtanen, T. (2016b). Tut database for acoustic scene classification and sound event detection. In *24th European Signal Processing Conference (EUSIPCO)*, pages 1128–1132. IEEE.

Mesaros, A., Heittola, T., and Virtanen, T. (2018). A multi-device dataset for urban acoustic scene classification. *arXiv preprint arXiv:1807.09840*.

Mobahi, H., Collobert, R., and Weston, J. (2009). Deep learning from temporal coherence in video. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 737–744. ACM.

Mohar, B., Alavi, Y., Chartrand, G., and Oellermann, O. (1991). The Laplacian spectrum of graphs. *Graph theory, combinatorics, and applications*, 2(871-898):12.

Mulimani, M., Jahnavi, U., and Koolagudi, S. G. (2017). Acoustic event classification using graph signals. In *Region 10 Conference (TENCON)*, pages 1812–1816. IEEE.

Mulimani, M. and Koolagudi, S. G. (2018). Robust Acoustic Event Classification using Bag-of-Visual-Words. In *INTERSPEECH*, pages 3319–3322.

Mulimani, M. and Koolagudi, S. G. (2019a). Extraction of MapReduce-based features from spectrograms for audio-based surveillance. *Digital Signal Processing*, 87:1–9.

Mulimani, M. and Koolagudi, S. G. (2019b). Robust acoustic event classification using fusion fisher vector features. *Apllied Acoustics*, 155:130–138.

Mulimani, M. and Koolagudi, S. G. (2019c). Segmentation and characterization of acoustic event spectrograms using singular value decomposition. *Expert Systems with Applications*, 120:413–425.

Mun, S., Shon, S., Kim, W., Han, D. K., and Ko, H. (2017). Deep neural network based learning and transferring mid-level audio features for acoustic scene classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 796–800. IEEE.

Nakamura, S., Hiyane, K., Asano, F., Nishiura, T., and Yamada, T. (2000). Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition.

Oppenheim, A. V. (1970). Speech spectrograms using the Fast Fourier Transform. *IEEE spectrum*, 8(7):57–62.

Pancoast, S. and Akbacak, M. (2012). Bag-of-audio-words approach for multimedia event classification. In *Thirteenth Annual Conference of the International Speech Communication Association*, pages 2105–2108.

Pandey, A. and Wang, D. (2019). A new framework for cnn based speech enhancement in the time domain. *IEEE/ACM Transactions on Audio, Speech, and Language Processing.*

Parascandolo, G., Huttunen, H., and Virtanen, T. (2016). Recurrent neural networks for polyphonic sound event detection in real life recordings. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6440–6444. IEEE.

Patterson, R. D., Robinson, K., Holdsworth, J., McKeown, D., Zhang, C., and Allerhand, M. (1992). Complex sounds and auditory images. In *Auditory physiology and perception*, pages 429–446. Elsevier.

Peng, Y.-T., Lin, C.-Y., Sun, M.-T., and Tsai, K.-C. (2009). Healthcare audio event classification using hidden markov models and hierarchical hidden markov models. In *2009 IEEE International Conference on Multimedia and Expo*, pages 1218–1221. IEEE.

Perperis, T., Giannakopoulos, T., Makris, A., Kosmopoulos, D. I., Tsekeridou, S., Perantonis, S. J., and Theodoridis, S. (2011). Multimodal and ontology-based fusion approaches of audio and visual processing for violence detection in movies. *Expert Systems with Applications*, 38(11):14102–14116.

Perronnin, F., Sánchez, J., and Mensink, T. (2010). Improving the fisher kernel for large-scale image classification. In *European conference on computer vision*, pages 143–156. Springer.

Phan, H., Hertel, L., Maass, M., Mazur, R., and Mertins, A. (2016a). Learning representations for nonspeech audio events through their similarities to speech patterns. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):807–822.

Phan, H., Maass, M., Hertel, L., Mazur, R., McLoughlin, I., and Mertins, A. (2016b). Learning compact structural representations for audio events using regressor banks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 211–215. IEEE.

Piczak, K. J. (2015a). Environmental sound classification with convolutional neural networks. In *25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE.

Piczak, K. J. (2015b). ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press.

Plinge, A., Grzeszick, R., and Fink, G. A. (2014). A bag-of-features approach to acoustic event detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3704–3708. IEEE.

Radhakrishnan, R., Divakaran, A., and Smaragdis, P. (2005). Systematic acquisition of audio classes for elevator surveillance. In *Proceedings of Society of Photo-Optical Instrumentation Engineers (SPIE)*, volume 5685, pages 64–71.

Raj, B. and Stern, R. M. (2005). Missing-feature approaches in speech recognition. *IEEE Signal Processing Magazine*, 22(5):101–116.

Rakotomamonjy, A. and Gasso, G. (2015). Histogram of gradients of time–frequency representations for audio scene classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(1):142–153.

Rattanaopas, K. and Kaewkeeree, S. (2017). Improving Hadoop MapReduce performance with data compression: A study using wordcount job. In *14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 564–567. IEEE.

Räty, T. D. (2010). Survey on contemporary remote surveillance systems for public safety. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(5):493–515.

Salamon, J. and Bello, J. P. (2017). Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283.

Salamon, J., Jacoby, C., and Bello, J. P. (2014). A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1041–1044. ACM.

Sánchez, J., Perronnin, F., Mensink, T., and Verbeek, J. (2013). Image classification with the Fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245.

Sandryhaila, A. and Moura, J. M. (2014). Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. *IEEE Signal Processing Magazine*, 31(5):80–90.

Schilit, B. N., Adams, N., Want, R., et al. (1994). Context-aware computing applications. In *First Workshop on Mobile Computing Systems and Applications*, pages 85–90.

Schmitt, M., Ringeval, F., and Schuller, B. W. (2016). At the border of acoustics and linguistics: Bag-of-audio-words for the recognition of emotions in speech. In *INTERSPEECH*, pages 495–499.

Schröder, J., Goetze, S., and Anemüller, J. (2015). Spectro-temporal gabor filterbank features for acoustic event detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(12):2198–2208.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Sharan, R. V. and Moir, T. J. (2015). Subband time-frequency image texture features for robust audio surveillance. *IEEE Transactions on Information Forensics and Security*, 10(12):2605–2615.

Sharan, R. V. and Moir, T. J. (2018). Pseudo-color cochleagram image feature and sequential feature selection for robust acoustic event recognition. *Applied Acoustics*, 140:198–204.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905.

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98.

Shvachko, K., Kuang, H., Radia, S., and Chansler, R. (2010). The hadoop distributed file system. In *IEEE 26th symposium on Mass storage systems and technologies (MSST)*, pages 1–10. IEEE.

Slaney, M. et al. (1993). An efficient implementation of the patterson-holdsworth auditory filter bank. *Apple Computer, Perception Group, Technical Report*, 35(8).

Smith, N. and Gales, M. J. (2002). Using svms and discriminative models for speech recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 77–80. IEEE.

Somervuo, P., Harma, A., and Fagerlund, S. (2006). Parametric representations of bird sounds for automatic species recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(6):2252–2263.

Spielman, D. A. (2010). Algorithms, graph theory, and linear equations in Laplacian matrices. In *Proceedings of the international congress of mathematicians*, volume 4, pages 2698–2722.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Stevens, S. S., Volkmann, J., and Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190.

Stork, J. A., Spinello, L., Silva, J., and Arras, K. O. (2012). Audio-based human activity recognition using non-markovian ensemble voting. In *21st IEEE International Symposium on Robot and Human Interactive Communication*, pages 509–514. IEEE.

Stowell, D., Giannoulis, D., Benetos, E., Lagrange, M., and Plumbley, M. D. (2015). Detection and classification of acoustic scenes and events. *IEEE Transactions on Multimedia*, 17(10):1733–1746.

Tang, J., Sun, J., Wang, C., and Yang, Z. (2009). Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 807–816. ACM.

Temko, A., Malkin, R., Zieger, C., Macho, D., Nadeu, C., and Omologo, M. (2006a). Acoustic event detection and classification in smart-room environments: Evaluation of chil project systems. *Cough*, 65(48):5–11.

Temko, A., Monte, E., and Nadeu, C. (2006b). Comparison of sequence discriminant support vector machines for acoustic event classification. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 5, pages 721–724. IEEE.

Temko, A. and Nadeu, C. (2006). Classification of acoustic events using svm-based clustering schemes. *Pattern Recognition*, 39(4):682–694.

Temko, A. and Nadeu, C. (2009). Acoustic event detection in meeting-room environments. *Pattern Recognition Letters*, 30(14):1281–1288.

Unoki, M. and Akagi, M. (1999). A method of signal extraction from noisy signal based on auditory scene analysis. *Speech Communication*, 27(3-4):261–279.

Valero, X. and Alias, F. (2012). Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification. *IEEE Transactions on Multimedia*, 14(6):1684–1689.

Van Loan, C. F. (1976). Generalizing the singular value decomposition. *SIAM Journal on Numerical Analysis*, 13(1):76–83.

Varga, A. and Steeneken, H. J. (1993). 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*, 12(3):247–251.

Virtanen, T., Mesaros, A., Heittola, T., Plumbley, M., Foster, P., Benetos, E., and Lagrange, M. (2016). Proceedings of the detection and classification of acoustic scenes and events 2016 workshop (dcase2016), budapest, hungary.

Wall, M. E., Rechtsteiner, A., and Rocha, L. M. (2003). Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer.

Wan, V. and Renals, S. (2005). Speaker verification using sequence discriminant support vector machines. *IEEE Transactions on Speech and Audio Processing*, 13(2):203–210.

Wang, C., Yang, J., Xie, L., and Yuan, J. (2019). Kervolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 31–40.

Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., and Gong, Y. (2010). Locality-constrained linear coding for image classification. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3360–3367. IEEE.

Wang, X., Tu, Z., and Zhang, M. (2018). Incorporating statistical machine translation word knowledge into neural machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(12):2255–2266.

White, T. (2012). *Hadoop: The definitive guide.* O'Reilly Media, Inc.

Wu, X., Zhu, X., Wu, G.-Q., and Ding, W. (2014). Data mining with big data. *IEEE transactions on knowledge and data engineering*, 26(1):97–107.

Xiang, J., McKinney, M. F., Fitz, K., and Zhang, T. (2010). Evaluation of sound classification algorithms for hearing aid applications. In *2010 IEEE international conference on acoustics, speech and signal processing*, pages 185–188. IEEE.

Yang, J., Jiang, Y.-G., Hauptmann, A. G., and Ngo, C.-W. (2007). Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the international workshop on multimedia information retrieval*, pages 197–206. ACM.

Yang, J., Yu, K., Gong, Y., and Huang, T. (2009). Linear spatial pyramid matching using sparse coding for image classification. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1794–1801. IEEE.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., et al. (2009). The htk book (for htk version 3.4. 1), cambridge university. *Engineering Department*.

Zhang, K. and Chen, X.-W. (2014). Large-scale deep belief nets with mapreduce. *IEEE Access*, 2:395–403.

Zhang, T., Zheng, W., Cui, Z., Zong, Y., Yan, J., and Yan, K. (2016). A deep neural network-driven feature learning method for multi-view facial expression recognition. *IEEE Transactions on Multimedia*, 18(12):2528–2536.

Zhang, Z. and Schuller, B. (2012). Semi-supervised learning helps in sound event classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 333–336. IEEE.

Zhuang, X., Zhou, X., Hasegawa-Johnson, M. A., and Huang, T. S. (2010). Real-world acoustic event detection. *Pattern Recognition Letters*, 31(12):1543–1551.

List of Publications

Journal Publications

1. Manjunath Mulimani and Shashidhar G. Koolagudi (2019). Extraction of MapReduce-based features from spectrograms for audio-based surveillance. Digital Signal Processing, 87:1-9.
2. Manjunath Mulimani and Shashidhar G. Koolagudi (2019). Segmentation and characterization of acoustic event spectrograms using singular value decomposition. Expert Systems with Applications, 120:413-425.
3. Manjunath Mulimani and Shashidhar G. Koolagudi (2019). Robust acoustic event classification using Fusion Fisher Vector features. Apllied Acoustics, 155:130-138.

Conference Publications

1. Manjunath Mulimani and Shashidhar G. Koolagudi (2019). Locality-constrained Linear Coding based Fused Visual Features for Robust Acoustic Event Classification. In INTERSPEECH, pages 2558-2562
2. Manjunath Mulimani and Shashidhar G. Koolagudi (2018). Robust Acoustic Event Classification using Bag-of-Visual-Words. In INTERSPEECH, pages 3319-3322.
3. Manjunath Mulimani and Shashidhar G. Koolagudi (2018). Acoustic Event Classification using Spectrogram Features. In Region 10 Conference (TENCON), pages 1460-1464. IEEE.
4. Manjunath Mulimani, Jahnavi U. and Shashidhar G. Koolagudi (2017). Acoustic event classification using graph signals. In Region 10 Conference (TENCON), pages 1812-1816. IEEE.

Technical report

1. Manjunath Mulimani and Shashidhar G. Koolagudi (2016). Acoustic Scene Classification using MFCC and MP features. In IEEE AASP challenge on Detection and Classification of Acoustic Scenes and Events.

Brief Bio-Data

Personal Details

Name - Manjunath Mulimani

Date of Birth - 06th June 1989

Work Address

Manjunath Mulimani
Research Scholar, Department of CSE,
NITK Surathkal, Mangalore,
Karnataka, 575 025.
Email: manjunath.gec@gmail.com

Permanent Address

Manjunath Mulimani
D. No. 322, Tungal (Post),
Jamkhandi (Ta.), Bagalkot (Dist.),
Karnataka, 587 330.
Phone No: +91 (974) 207 3368

Qualification

M.Tech in Computer Science and Engineering from Visvesvaraya Technological University, Belagavi, Karnataka, India (2011-2013)

BE in Computer Science and Engineering from Visvesvaraya Technological University, Belagavi, Karnataka, India (2007-2011)

Previous Work experience

Worked as an Assistant Professor in Sahyadri College of Engineering and Management, Mangaluru, Karnataka (2013-2015)