ANKARA YILDIRIM BEYAZIT UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES



DETECTION AND RECOGNITION OF SOUNDS FROM HAZARDOUS EVENTS FOR SURVEILLANCE APPLICATIONS

Ph.D. Thesis by

Yüksel ARSLAN

Department of Electrical and Computer Engineering

December, 2018

ANKARA

DETECTION AND RECOGNITION OF SOUNDS FROM HAZARDOUS EVENTS FOR SURVEILLANCE APPLICATIONS

A Thesis Submitted to

The Graduate School of Natural and Applied Sciences of

Ankara Yıldırım Beyazıt University

In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Electrical and Computer, Department of Electrical and Computer Engineering

by

Yüksel ARSLAN

December, 2018

ANKARA

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"DETECTION AND RECOGNITION OF SOUNDS FROM HAZARDOUS EVENTS FOR SURVEILLANCE APPLICATIONS** under the supervision of **PROF. DR. HÜSEYİN CANBOLAT** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Hüseyin CANBOLAT

Supervisor

Assoc. Prof. Dr. Asaf Behzat ŞAHİN

Jury Member

Asst. Prof. Dr. Baran USLU

Jury Member

Asst. Prof. Dr. Ömer KARAL

Jury Member

Asst. Prof. Dr. Hakan TORA

Jury Member

Prof. Dr. Ergün ERASLAN

Director

Graduate School of Natural and Applied Sciences

I hereby declare that, in this thesis which has been prepared in accordance with the Thesis Writing Manual of Graduate School of Natural and Applied Sciences,

- All data, information and documents are obtained in the framework of academic and ethical rules,
- All information, documents and assessments are presented in accordance with scientific ethics and morals,
- All the materials that have been utilized are fully cited and referenced,
- No change has been made on the utilized materials,
- All the works presented are original,

and in any contrary case of above statements, I accept to renounce all my legal rights.

Date: **Signature :……**

Name & Surname:…….

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my supervisor, Prof. Dr. Hüseyin CANBOLAT for his support and motivation during my study. His guidance assisted me all the time of my research and while writing this thesis.

I am deeply thankful to Asst. Prof. Baran USLU for his support during paper preparation and real-time tests. His friendly way of support helped me to continue and finish the thesis.

2018, December **Yüksel ARSLAN**

DETECTION AND RECOGNITION OF SOUNDS FROM HAZARDOUS EVENTS FOR SURVEILLANCE APPLICATIONS

ABSTRACT

In this thesis we propose a system which uses algorithms for detection and recognition of impulsive sounds namely gunshots, screaming and traffic accidents. Such a system can be regarded as Environmental Sound Recognition System (ESR). We proposed two system designs. First system design has three subsystems and second system has two subsystems. Subsystems are sound database, impulsive sound detection and impulsive sound recognition subsystems. Second system design combines second and third as one subsystem. We designed and implemented a relational database model for storing impulsive sounds first then extended it to store all environmental sounds.

In the impulsive sound detection subsystem, we proposed to use impulsive noise detection algorithms for the detection of impulsive sounds. We implemented some algorithms and showed Warped Linear Prediction (WLP) outperforms an energy based algorithm which is mostly used in literature for impulsive sound detection, regarding false positives and miss detections.

In the recognition subsystem we employed Deep Neural Networks (DNN) which is state of the art. We developed three models for the recognition of gunshot, scream and traffic accident. For the performance evaluation we made offline and online tests. Online tests revealed that the reverberation degradation effect on recognition which was well known for distant speaker recognition. Our DNN models outperforms earlier research made on the same datasets.

At the end of the thesis we mention emerging sound recognition applications and later deeply investigate audio integration to video surveillance networks to improve the performance of this systems.

Keywords: Impulsive Sound, Sound Event Detection, Sound Event Recognition, Machine Learning, Deep Neural Networks, Audio Signal Processing.

GÖZETLEME AMAÇLI UYGULAMALAR İÇİN TEHLİKELİ OLAYLARDAN GELEN SESLERİN TESPİTİ VE TANINMASI

ÖZ

Bu tezde, silah sesi, çığlık ve araba kazası gibi darbesel seslerin tespiti ve tanınmasını yapacak bir system geliştirilmesi amaçlanmaktadır. Böyle bir system Çevresel Ses Tanıma Sistemi olarak adlandırılabilir (ÇST). İki system tasarımı önerilmiştir. Birinci sistem tasarımında üç adet ikinci system tasarımında iki adet alt system bulunmaktadır. Alt sistemler ses veritabanı, darbesel ses tespiti ve darbesel ses tanıma alt sistemleridir. İkinci sistem tasarımı iki ve üçüncü altsistemleri bir alt system altında toplar. İlk önce darbesel sesleri depolayacak bir ilişkisel veritabanı modeli tasarımı yapıp bunu gerçekleştirdik. Daha sonra bu tasarımı tüm çeresel sesleri depolayacak şekilde genişlettik.

Darbesel ses tespit alt sisteminde darbesel ses tespiti amacıyla darbesel gürültü tespit algoritmalarının kullanılmasını önerdik. Bazı algoritmaları gerçekleştirdik ve sonuçlar göstermiştir ki; Warped Linear Prediction (WLP) literatürde en çok kullanılan bir enerji tabanlı algoritmayı yanlış tespit ve tespit kaçırma bakımından geride bırakmıştır.

Tanıma alt sisteminde son zamanlarda en iyi sonucu veren Derin Sinir Ağları (DSA) kullanılmıştır. Silah sesi, çığlık ve araba kazası tanıması için üç DSA modeli geliştirilmiştir. Performans değerlendirmesi için offline ve online testler yapılmıştır. Online testlerde uzak konuşmacı tanıması alanında çok iyi bilinen reverberasyonun tanıma üzerindeki bozma etkisi görülmüştür. DSA modellerimiz aynı veri setleri üzerinde daha önce yapılan çalışmalardan daha iyi sonuç vermiştir.

Tezin sonunda, son zamanlarda ortaya çıkmakta olan ses tanıma uygulamaları anlatılmakta ve mevcut video güvenlik sistemlerinin performansının iyileştirilmesi için ses entegrasyonu yapılması detaylı olarak incelenmektedir.

Anahtar Kelimeler: Darbesel ses, ses olayı tespiti, ses olayı tanıma, makine öğrenmesi, derin sinir ağları, ses sinyali işleme.

CONTENTS

Ph.D. THESIS EXAMINATION RESULT FORM ii
ACKNOWLEDGMENTS iv
ABSTRACT v
ÖZ vii
NOMENCLATURE xi
LIST OF FIGURES xiii
LIST OF TABLES xvi

CHAPTER 1 1
INTRODUCTION 1
1.1 Motivation 1
1.2 Contributions 2
1.3 Organization of This Document 3

CHAPTER 2 4
OVERVIEW OF THE PROPOSED SYSTEM 4
2.1 Review of ESR Systems 4
2.2 ESR Application Areas 5
2.3 System Architecture 8
2.4 Development and Test Architecture 10
2.5 Deployment Architecture 11

CHAPTER 3 12
SOUND LIBRARY DEVELOPMENT 12
3.1 Sound Database Work in ESR Systems 12
3.2 Review of ESR Libraries 14
3.3 Database Design 17
3.3.1 Functional Requirements 17
3.3.2 Non-Functional Requirements 18
3.3.3 Implementation of the Database 19

3.4 Database Usage and Tests 20
3.5 Data for the Database 21
3.6 Database Extension 21
3.6.1 Taxonomy of Environmental Sounds 22
3.6.2 Non-Functional Requirements 23
3.6.3 Functional Requirements 24
3.6.4 Graphical User Interface (GUI) Requirements 24
3.6.5 Database Implementation 25

CHAPTER 4 28
IMPULSIVE SOUND DETECTION 28
4.1 Introduction 28
4.2 Classification of Impulsive Sound/Noise Detection Algorithms 30
4.3 Online Impulsive Sound Detection Algorithms 31
4.3.1 Algorithms based on signal energy 31
4.2.2 Algorithms based on sound modeling 32
4.2.3 Detection by classification 32
4.4 Online Impulsive Noise Detection Algorithms 34
4.4.1 Time domain methods 34
4.4.2 Sound modeling based 34
4.5 Offline Impulsive Sound Detection Algorithms 35
4.6 Offline Impulsive Noise Detection Algorithms 36
4.6.1. Algorithms based on sound modeling 36
4.6.2. Other detection algorithms 36
4.7 Impulsive Sound Detection Methods 37
4.8 Implementation of Impulsive Sound/Noise Detection Algorithms 38
4.8.1 Energy Based Algorithms 38
4.8.2 Prediction Based Algorithms 41
4.8.3 Detection Using Discrete Wavelet Transform (DWT) 42
4.9 Impulse Detection Algorithm Performances 43
4.10 Discussion 45

CHAPTER 5 47
IMPULSIVE SOUND RECOGNITION 47

5.1 Machine Learning 47
5.2 Cost Function 49
5.3 Gradient Descent Algorithm 49
5.4 Deep Neural Networks 49
5.5 Implementation of DNN for This Thesis 51
5.6 Recognition of Gunshot, Scream and Car Crash with DNN 51
5.6.1 Data, Feature Extraction and Model Preparation 52
5.6.2 Testing 56

CHAPTER 6 64
EMERGING AUDIO SURVEILLANCE APPLICATIONS 64
6.1 Emerging Applications 64
6.2 Integration of Audio Modality to Video Surveillance Networks for Open Area/Campus Security 67
6.2.1 Introduction 67
6.2.2 Latest Trends in Automatic Video Surveillance Systems 70
6.2.3 Integration of Audio to Video Surveillance Network 73

CHAPTER 7 81
CONCLUSION AND FUTURE WORK 81
7.1 Summary of Completed Work 81
7.2 Future Work 82

REFERENCES 83

CURRICULUM VITAE 96

NOMENCLATURE

AESR	Automatic Environmental Sound Recognition
AR	Auto Regressive
ARIMA	Auto Regressive Integrated Moving Average
ARMA	Auto Regressive Moving Average
APD	Absolute Predictive Deviation
ASR	Automatic Speech Recognition
CASA	Computational Auditory Scene Analysis
CASR	Computational Audio Scene Recognition
CCTV	Closed Circuit TV
D	Deletions
DBN	Deep Belief Network
DCASE	Detection and Classification of Acoustic Scenes and Events
DET	Detection Error Tradeoff
DNN	Deep Neural Network
DSR	Distant Speaker Recognition
EER	Equal Error Rate
ER	Error Rate
ESC	Environmental Sound Classification
ESR	Environmental Sound Recognition
FN	False Negative
FP	False Positive
GMM	Gaussian Mixture Model
GUI	Graphical User Interface
IDC	Intelligent Data Corporation
IP	Internet Protocol
LAN	Local Area Network
LMS	Least Mean Square
LPC	Linear Predictive Coding
MD	Miss Detection
ML	Machine Learning
MFCC	Mel Frequency Cepstral Coefficients

PTZ	Pan-Tilt-Zoom
RelU	Rectified Liner Unit
RMS	Root Mean Square
RMSE	Root Mean Square Error
RWCP	Real World Computing Partnership
S	Substitutions
SED	Sound Event Detection
SER	Sound Event Recognition
SNR	Signal to Noise Ratio
STFT	Short Time Fourier Transform
SVM	Support Vector Machine
TCP	Transport Control Protocol
TK	Teager-Kaiser
TN	True Negative
TP	True Positive
TPR	True Positive Rate
UDP	User Datagram Protocol
VSaaS	Video Surveillance as a Service
WLP	Warped Linear Prediction
xGSSs	x. Generation Surveillance System

LIST OF FIGURES

Figure 2.1 The block diagram of the system to be developed..... 8

Figure 2.2 Block diagram of the system to be developed second approach 9

Figure 2.3 Pilot system for development and testing of the detection and recognition algorithms..... 10

Figure 3.1 ESR system in training phase 12

Figure 3.2 ESR system in testing phase 13

Figure 3.3 ESR system in production phase 13

Figure 3.4 RWCP dry source sound clips structure 15

Figure 3.5 DCASE 2017 task-2 development dataset “yaml” file for glass break ... 16

Figure 3.6 CSV file explaining Urbansound database 17

Figure 3.7 Database table diagram..... 19

Figure 3.8 Workflows for database testing 21

Figure 3.9 Grouping of sounds based on sound producing events 22

Figure 3.10 Grouping of sounds defined in [54]..... 22

Figure 3.11 A part of the urban sound taxonomy defined in [52]..... 23

Figure 3.12 Environmental sounds grouping in the database 23

Figure 3.13 Environmental sounds table diagram..... 26

Figure 3.14 Desktop application GUI for administrators to data load..... 27

Figure 3.15 WEB GUI for end user tools 27

Figure 4.1 Graphs of (a)gunshot (b)traffic accident (c)scream 29

Figure 4.2 A typical ESR system [56] 29

Figure 4.3 Taxonomy of Impulsive Sound/Noise Detection Algorithms and Our New Approach for Impulsive Sound Detection 30

Figure 4.4 Hazardous Sound Event Detection Method 1 37

Figure 4.5 Hazardous Sound Event Detection Method 2 38

Figure 4.6 Hazardous Sound Event Detection Method 3 38

Figure 4.7 Framing of incoming samples 39

Figure 4.8 Power calculation 39

Figure 4.9 Powers are windowed and normilized window size = 30 40

Figure 4.10 DET curves of energy based algorithm (base algorithm), LPC and WLP for detection of gunshot for different EBR values. (a) EBR = -6, (b) EBR = 0, (c) EBR = 6 44

Figure 4.11 DET curves of energy based algorithm (base algorithm), LPC and WLP for detection of glass break for different EBR values. (a) EBR = -6, (b) EBR = 0, (c) EBR = 6 44

Figure 4.12 TP rate of algorithms for traffic and crowd noise type 45

Figure 4.13 FP of algorithms for traffic and crowd noise type 45

Figure 5.1 Machine learning training 47

Figure 5.2 Machine learning testing 48

Figure 5.3 Machine learning in usage 48

Figure 5.4 (a) Logistic regression (b) 1 Hidden layer NN (c) 2 Hidden layers (d) 5 Hidden layer-Deep NN 50

Figure 5.5 A DNN node and the calculation in it 51

Figure 5.6 Feature Extraction 55

Figure 5.7 DNN model for car crash 56

Figure 5.8 Offline SED 58

Figure 5.9 Online test for correct detection in room environment 61

Figure 5.10 Online testing environment at Atılım University 61

Figure 6.1 Audio surveillance for home/campus security 64

Figure 6.2 E-Call system 66

Figure 6.3 Proposed E-Call system for old cars 66

Figure 6.4 General video surveillance network 68

Figure 6.5 Control room of video surveillance network 69

Figure 6.6 Audio integrated general system 75

Figure 6.7 Process graph for audio analysis system development [93]. 76

Figure 6.8 Initial system design 78

LIST OF TABLES

Table 5.1 Gunshot dataset 53

Table 5.2 Car crash dataset..... 54

Table 5.3 Scream dataset..... 54

Table 5.4 Event sound feature properties 55

Table 5.5 DNN hyper-parameters 56

Table 5.6 Definitions of metrics used in SED [95] 57

Table 5.7 Gunshot detection performance..... 59

Table 5.8 Comparison of gunshot detection with the baseline and some participants of DCASE 2017 Task-2 challange on the development dataset 59

Table 5.9 Performance of scream detection 60

Table 5.10 Accuracy comparison of secream detection with proposed method and other two studies on the same dataset 60

Table 5.11 Performance of car crash detection 60

Table 5.12 Accuracy comparison of car crash detection with proposed method and the work in [35] on the same dataset..... 60

Table 5.13 Comparison of offline and online scream detection..... 62

Table 5.14 Comparison of offline and online carcrash detection..... 62

Table 6.1 System requirements 75

CHAPTER 1

INTRODUCTION

Traditionally audio recognition focused on music and speech signals. Automatic Environmental Sound Recognition (AESR) has received more attention in recent years. ESR can find application in areas such as acoustic surveillance, bio-acoustical monitoring, environmental context detection, healthcare applications. ESR can also be used for audio search applications on Internet and for robot navigation and more. The term Sound Event Recognition (SER) and Sound Event Detection (SED) have the same meaning as ESR. In this thesis we will use ESR or SED.

In this thesis ESR is used for surveillance applications for security in general. Our objective is to develop a system to detect gunshots, human screams and traffic accidents in real time.

1.1 Motivation

Impulsive sounds such as gunshots, explosions, screams etc. can be regarded as a subset of environmental sounds. Automatic detection of such sounds can be used in surveillance applications:

- Detection and recognition of bell ringing, siren or car horn can help elderly or disabled persons affected in their hearing capabilities.
- Detection of gunshots and direction/position of snipers can be helpful in military.
- Glass breaking, siren detection can be used together with video surveillance for more accurate alarm identification.
- Detection of gunshots, human screaming, car crash can help in city security forces.

Our main objective in this thesis is to develop a system at the end which will be used together with in city surveillance cameras to detect sounds such as gunshots, screaming and traffic accidents.

1.2 Contributions

Proposed system consists of three main subsystems. These are sound library subsystem, impulsive sound detection subsystem and impulsive sound recognition subsystem.

- Sound library subsystem; consists of the sounds which will be detected and recognized.
- Impulsive sound detection subsystem; will inspect coming sound signal frame by frame and detect start of the desired hazardous event and pass the frame to recognition subsystem.
- Impulsive sound recognition subsystem; will take the frame and inspect if it consists of hazardous event or not.

We intend to develop a system which will run in real time. Traditional ESR work just focuses on developing algorithms. We will develop algorithms and use it in a real time application. A sound database will initially be developed as a sound library subsystem. We will have two contributions in the database development. First it will be based on relational database system and secondly it will be used after the system is operational, for online learning [1] [2].

The detection stage will be fast enough to cope with real time incidents. Algorithms used in impulsive sound detection and impulsive noise detection will be gathered. It will be the first research comparing impulsive sound and impulsive noise detection algorithms.

At the recognition stage we will use state of the art algorithms namely Deep Neural Networks (DNN). DNNs are currently the state of art for Machine Learning (ML).

After developing detection and recognition algorithms, another important subject to solve is the deployment of the applications. For these algorithms to be useful for surveillance applications we should think how to deal with many sounds from many microphones installed different locations. We will develop a networking and scalable application platform to deal with this problem. This deployment model is used together with surveillance camera systems to improve the effectiveness of video camera surveillance systems.

1.3 Organization of This Document

After this Introduction section, the document continues as follows:

Chapter 2; explains the overview of the proposed system.

Chapter 3; explains sound library.

Chapter 4; explains detection system.

Chapter 5; explains recognition system.

Chapter 6; explains emerging audio surveillance applications.

Chapter 7; explains conclusion and future work.

CHAPTER 2

OVERVIEW OF THE PROPOSED SYSTEM

2.1 Review of ESR Systems

There is not much work in literature on real/non-real systems of automatic ESR for practical applications. There are a lot of works about ESR but many of them just exploit the some features and machine learning algorithms using these features to recognize some kinds of sounds working on offline (recorded) data. A complete system running on online (real-time) data is very rare. In [3] a system is proposed which consists of detection and recognition stages. But it is not evident from the work how this system works. In [4] a complete system is designed as part of INDECT project [5]. In [6] a parallel architecture is explained for hazardous sound event recognition. In [7] a system is designed using Android OS for ESR. In this thesis, detection and recognition algorithms and their performances will be exploited on offline and real-time data. At the end, we will design a system which will use these algorithms for audio surveillance. This system will integrate audio modality to currently used video surveillance system for more effective usage.

Historically non-speech sound recognition has not received as much attention as automatic speech recognition (ASR). ASR has well established algorithms and databases while research has begun much earlier. AESR is taking some algorithms from the field of ASR. There are other fields related to AESR beside ASR and we will explain them below:

Computational Auditory Scene Analysis:

Computational auditory scene analysis (CASA) has been an active area of research in the end of nineties [8]-[10]. It refers to the computational analysis of an acoustic environment, and the recognition of distinct sound sources and events in it. CASA has begun with the book of Bergman “Auditory Scene Analysis” [11]. The other

studies in this area can be found in [12] and [13]. In the thesis [14] another area, Computational Audio Scene Recognition (CASR) was studied. CASR is different from the CASA such that it does not recognize individual events in the scene; it tries to recognize the scene as a whole such as street.

Speech Recognition:

The task is to transcribe continuous speech to text, by classifying phonemes according to previously trained examples. Other aspects which contribute to the variability of speech are also considered interesting, such as speaking rate, emotion and accent [15].

Speaker Recognition:

The field contains several sub-topics including speaker verification for biometrics [16] and speaker diarisation, which is the task of identifying "who spoke when" [17]. It requires a model to be created which uniquely identifies the speech of an individual, which does not necessarily require the recognition of individual phonemes or words.

Language Identification:

This is commonly used for machine translation [18], and is similar to speaker recognition in that systems may model the variability of the individual languages, rather than recognize the spoken words.

2.2 ESR Application Areas

In military, forensic and law enforcement domain there are studies on gunshot detection systems. In [19], a gunshot detection system is proposed. The proposed system consists of two parts. First part is for detection of impulsive sounds and it is less costly. Second part is for recognition of gunshots and it is more complex. For the recognition stage a template matching method in conjunction with eight order LPC coefficients are used to train Support Vector Machine (SVM) algorithm.

In [20], an audio event detection system which automatically classifies an audio event as noise, scream or gunshot is proposed. The proposed classification system uses two Gaussian Mixture Models (GMMs) running in parallel. One GMM is used for scream and the other is used for gunshot. Each GMM uses different set of features selected from a set of 47 audio features.

In [21], the template matching algorithm for gunshot detection is evaluated to be used in noisy environments. This simple method is compared with other more complex methods

In [22], a two-stage system is proposed for detection and recognition of impulsive sounds. Detection is done by applying an adaptive threshold on the energy of input signal. Recognition stage uses neural network algorithms. It is used in a digital signal processing device designed to detect and recognize gunshots, glass breaking and sirens.

In [23], the gunshot blast is used to identify the caliber of the gun. Autocorrelation functions are calculated in the preprocessing stage. Fourier cosine coefficients are used as features. These features are compared with a reference to detect the caliber of the gun.

In [24] and [25] ESR is used for robot navigation

ESR can be used for home monitoring. It can be used to assist elderly people living in their home alone [26], [27].

In [28] it is used for home automation.

In [29] matching pursuit based algorithm is used to select most relevant features from a dictionary of atoms. This dictionary contains Gabor atoms. The selected Gabor atoms are used along with Mel Frequency Cepstral Coefficients (MFCC) features to recognize 14 sound classes. It is shown that if recognition algorithm is used with selected Gabor atoms, recognition rate is increasing significantly.

In [30] and [31], ESR is used for recognition of animal sounds.

Audio Surveillance:

Since the audio surveillance is the field of our thesis, we will pay more attention to this field and investigate the applications in this field deeply.

In [32], a survey of modern surveillance systems is made. Modern surveillance systems face two important problems:

- Video cameras are constantly being installed for security reasons in prisons, banks, automatic teller machines, petrol stations, and elevators, which are the most susceptible for criminal activities. Usually, the video camera is connected to a recorder or to a display screen from which security personnel constantly monitor suspicious activities. As security personnel typically monitor multiple locations simultaneously this manual task is labor intensive and inefficient. Significant stress may be placed on the security personnel involved [33].

- There are immediate needs for automated surveillance systems in commercial, military applications, and law enforcement. Mounting video cameras is inexpensive, but locating available human resources to survey the output is expensive. Despite the usage of surveillance cameras in banks, stores, and parking lots, video data currently are used only retrospectively as a forensic tool, thus losing its primary benefit as an active real-time medium. What required is a continuous 24-hour monitoring of surveillance video to alert security officers of a burglary in progress, or a suspicious individual lingering in a parking lot, while there still is time to prevent the criminal offence [34].

Integrating audio modality to current video surveillance systems and utilizing hazardous SED, the above problems can be eliminated significantly.

Other application fields for audio surveillance are as follows:

In [35], a system is designed for road surveillance by using audio surveillance. This system uses a bag of words approach for sound representation. SVM is used for classification.

In [36], a system is designed for surveillance of public transport. A method is explained for shout detection embedded in railway sound using GMM and SVM.

In [37], systematic acquisition of audio classes for elevator surveillance is explained. First the system detects outlier in time domain then automatically clusters these outliers. Classes of the clustered sound are determined and afterwards modelled by a GMM.

In [38], automatic surveillance of the acoustic activity in our living environment namely an office corridor. First events are detected and segmented by an FFT base detector. K-means algorithm is applied for the detected segments to determine the clusters. Manual classification is done.

In [39], talk, shout, footstep and knock are classified by GMM classifier by using a top-down approach. First events are classified as foreground or background then vocal or non-vocal. Vocal events are classified as talk and cry, non-vocal events are classified as knock and footsteps.

2.3 System Architecture

We plan to detect impulsive sounds and recognize these sounds using two approaches. Block diagram of first approach is in Figure 2.1. This approach takes into account that the hazardous sounds are impulsive sounds and detection of these sounds can be achieved faster than recognition. So it is necessary to first detect and then to recognize the hazardous sound events.

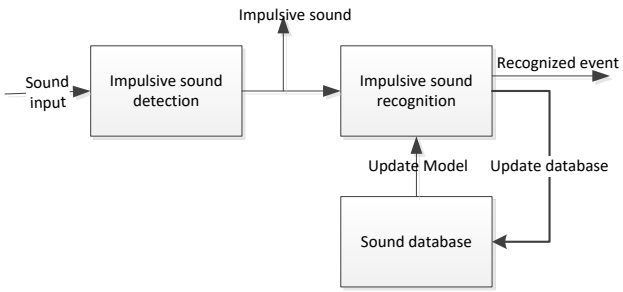


Figure 2.1 The block diagram of the system to be developed

Block diagram of second approach is in Figure 2.2. If we find algorithms fast and successful enough for recognition then we can omit the detection stage and use only recognition algorithms for detection and recognition of the hazardous sound events.

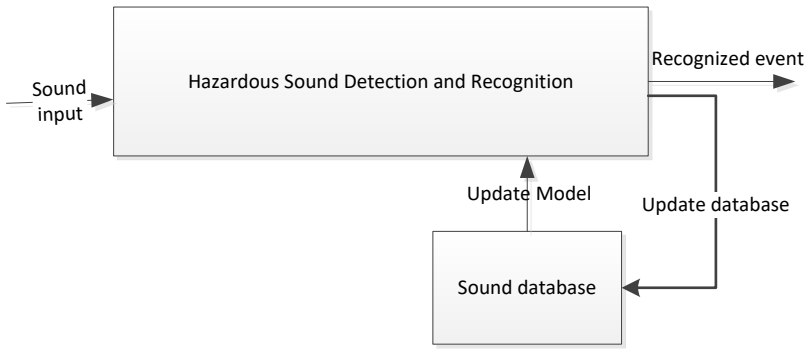


Figure 2.2 Block diagram of the system to be developed second approach

Sound library is the first subsystem that will be developed. Firstly it is necessary to build a library of the sounds which will be recognized. This library will comprise of the sounds such as scream, gunshot, siren, glass break and car horn. The list may be expanded later. We will try to build this library from Internet and maybe by way of our actual trials. This library will also consist of noises. These noises can be traffic, thunder, crowd, helicopter etc. Our detection and recognition algorithm will work in these conditions. Our success measure will be based on Signal to Noise Ratio (SNR).

An *impulsive sound detection* algorithm will be developed and tested. The developed algorithm will be compared to other algorithms found in the literature. The developed algorithm will be robust such that it can find impulsive sounds in highly noisy environments such as traffic.

At last an impulsive sound recognition subsystem will be developed. If an impulsive sound is detected then the recognition algorithm will classify this sound as scream, gunshot, car horn, siren or glass break. If nothing is matched, it is recorded to library for further processing.

In the first approach the impulsive sound detection subsystem works all the time, recognition subsystem works after detection.

2.4 Development and Test Architecture

We developed a pilot system to develop and test the detection and recognition algorithms at the beginning. The block diagram of this pilot development and test system is seen in Figure 2.3.

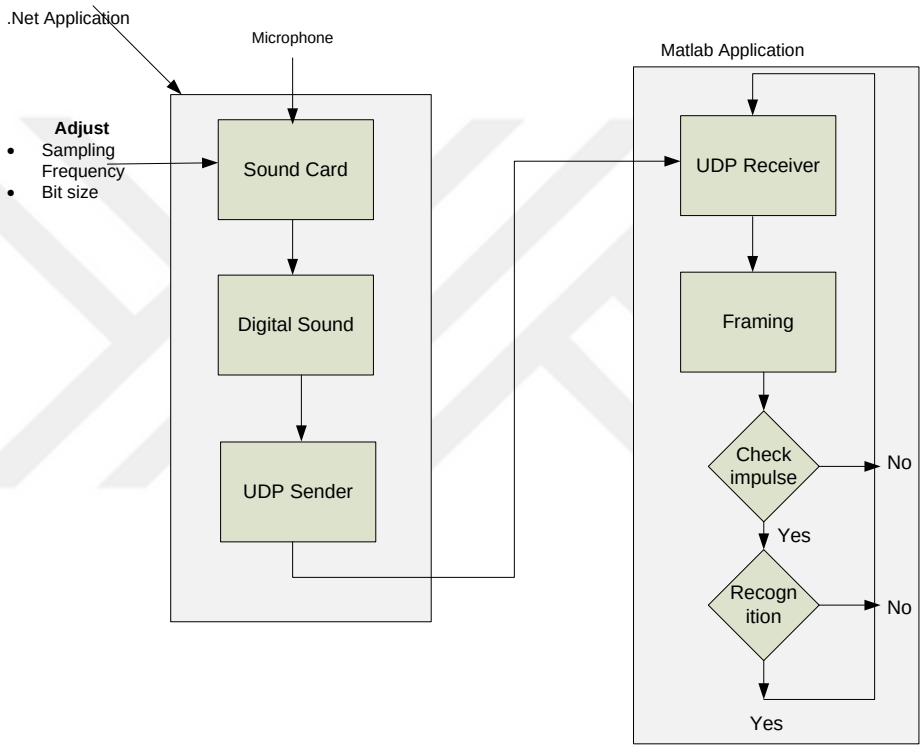


Figure 2.3 Pilot system for development and testing of the detection and recognition algorithms

This system has the following properties:

- It is divided into two applications. One application is listening to the environment with a microphone and digitizing the sound and then sending this sound over network using User Datagram Protocol (UDP) to the other application.

- This first application is a Windows application developed using Microsoft Visual Studio and C#.
- Second application receives the sound data coming over the network and decides the event type.
- The detection and the recognition algorithms run in the second application.
- Second application is developed using Matlab.
- This system is also suitable for our ultimate aim which is real time hazardous event detection and recognition.

2.5 Deployment Architecture

One of our motivations is to develop a system which will be used in cities along with video cameras to detect the incidents in real time. For this purpose we should consider the final deployment architecture at the beginning. Two architectures are possible, distributed and centralized.

Distributed deployment architecture: In this architecture, the microphones installed all over the city are connected to small computers. These microphones and the computers are independent processing nodes. All the detection and recognition algorithms run in these nodes and they just send the final decision to the center. These processing nodes should update a common database and their detection and recognition models should be updated from the center.

Centralized deployment architecture: In this architecture, the deployed microphones and processing nodes just send the sound signals over the network to the center. All processing activities of detection and recognition are done by the computers at the center. Scalability can be a problem for this type of architecture, but it is easy to maintain the database and update the models.

CHAPTER 3

SOUND LIBRARY DEVELOPMENT

It is necessary to establish a database which comprises the sounds to be recognized in advance to be able to recognize the environmental sounds. In this section we present the design and pilot establishment of a database which will become the base of the impulsive environmental sound recognition system. This database which comprises impulsive environmental sounds will provide the requested features for the detection and recognition of sounds obtained from the environment. As offline it will be used for development and test platform for algorithms that will be developed. Besides we explain the measures taken into account during data collection for the database, since it is hard to find and record impulsive environmental sounds. In the following section we will describe how a sound database is used in the development of an ESR system.

3.1 Sound Database Work in ESR Systems

ESR system design is started with the training phase. How a sound database is used in training phase of an ESR system is explained in Figure 3.1.

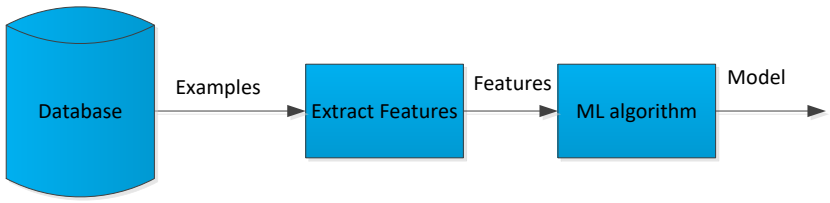


Figure 3.1 ESR system in training phase

During the development stage of an ESR system the desired machine learning (ML) algorithm must be trained with the sounds to be recognized. The database provides the sound clips to be recognized and other sound clips which are negative examples. After training, a model is developed and this model is used for testing. In Figure 3.1,

the database provides positive and negative examples; features are inputs to the ML algorithm and ML algorithm by using these features produce the model.

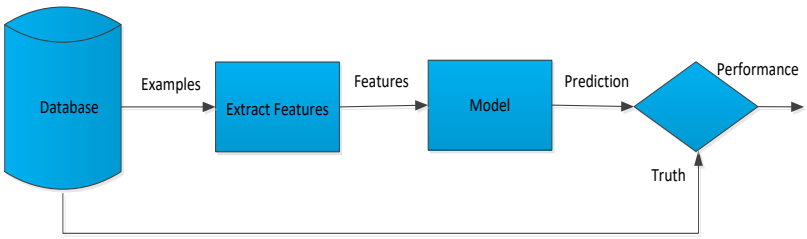


Figure 3.2 ESR system in testing phase

In Figure 3.2, testing phase of an ESR system is shown. In testing phase database provides the positive and negative examples, model produces the predictions about the examples provided and at last predictions are compared with the truth provided by the database and a performance rate is reached. According to this performance rate, ML algorithm or the feature set or other parameters of the ML algorithm may be needed to change. Then the whole training and testing phase start again.

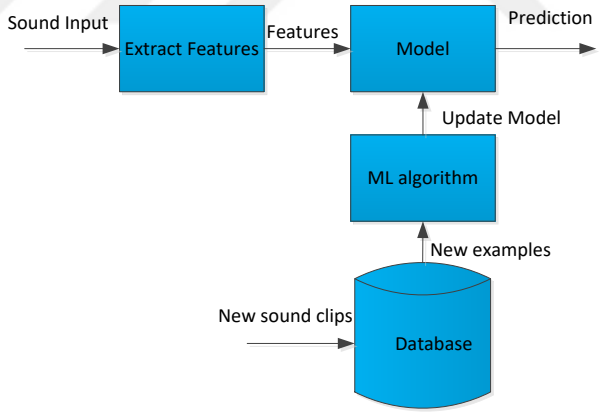


Figure 3.3 ESR system in production phase

After model creation and testing, this model is used in the production phase (Figure 3.3). In production phase, sounds come from the environment, model makes the predictions about the classes of sounds and we actually may not know the real truth.

In production stage we may update sounds in database, update the model even we may change the ML algorithm. In each case we need the database.

3.2 Review of ESR Libraries

In literature there are many papers written on ESR [40]-[44]. Although in these papers it is referenced to libraries or databases which consist of used sound types, it is given no information about details of the libraries. Only they give some information about their technical details. In the process of development of an ESR system, you must work on many sound clips, many features extracted from these clips and many algorithms which will use these features. After development of the system, if it would be open to development, it was needed to add new sound clip types, new sound clips of these types, new features and algorithms. To make all these easy, it would be very beneficial to develop a database system.

There are some sound databases developed by the research groups of universities which consist of the sounds created and recorded in laboratory. These databases are constructed for an aim or just to provide a source for other researchers. http://www.auditorylab.org is one of these. This database was constructed by Carnegie Mellon University to examine the human ability to use sound to understand what events are happening in the environment. All the sounds in the database are recorded in a controlled way. The laboratory and the recording media used are technically detailed. In the construction of database sounds are grouped by the event that makes the sound. Sounds of events like impact, rolling can be downloaded from this database.

Another work is http://www.desra.org. In [45] the aim of the database, details of the design and the sources used are explained. It is designed as multi-purpose database. The database contains variety of sounds from different events with thorough background information for each sound file. It is accessible from the Internet.

[46] explains the database accessed from http://www.daresounds.org/. Aim of this database is to collect everyday sounds that are recorded in the full complexity of

everyday situations and to make them available to the scientific community. DARES aims to be suitable for auditory perception research and as a validation platform of everyday sound recognition systems.

An ESR database required for a research in USA Army Research Laboratory is detailed in the report [47]. In this report it is given which properties of the sound should be documented, criteria for the sounds to be included into database and if recording is necessary, the criteria of recording is given.

One of the databases that have been mostly reverenced is Real World Computing Partnership's (RWCP) non-speech database [48]. Using a standard single microphone, acoustic signals of about 100 types of sound sources were measured in an anechoic room as the dry source. By using the impulse response of three kinds of rooms, convolution with the 48 kHz sampling signal was carried out to reconstruct the sound signal in respective rooms. The database structure (structure of directories) and navigation to sounds is explained and provided by an HTML page as shown in Figure 3.4 for dry sources. As an ESR developer we must examine the structure of HTML file and find the desired sound clips. Maintenance of the data and usage is not so easy with this kind of structure.

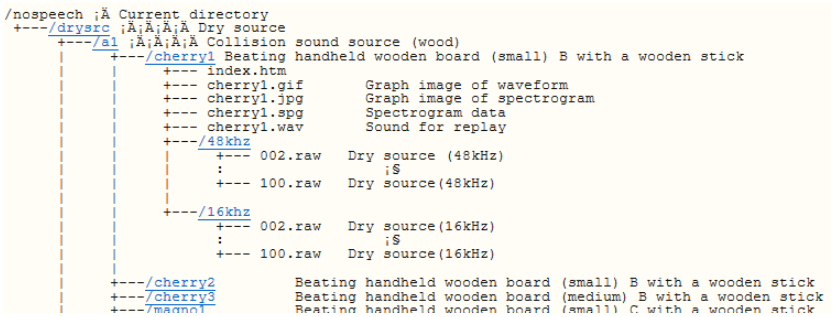


Figure 3.4 RWCP dry source sound clips structure

Detection and Classification of Acoustic Scenes and Events (DCASE) is an official IEEE Audio and Acoustic Signal Processing challenge. For challenge a database is prepared and it is also a resource for ESR researchers. DCASE 2016 challenge consists of four tasks [49]. The goal of Task-2 is to detect sound events for example

“bird singing”, “car passing by” that are present within an audio segment. To be prepared for the challenge two datasets, train and development are given. Train is used to create the model and development is to test the model. For training all sound files are in one directory and a readme file explains the details, such as sampling frequency, quantization bit depth, etc. For development dataset there is an annotation text file for each sound clip. Each .txt file contains information about the onset, offset, and event class for each event in the scene, separated by a tab.

DCASE 2017 Task-2 dataset contains “.yaml” files for annotations [50]. “.yaml” files (Figure 3.5) can be read by a Matlab command, so it is easy to work with these structured files.

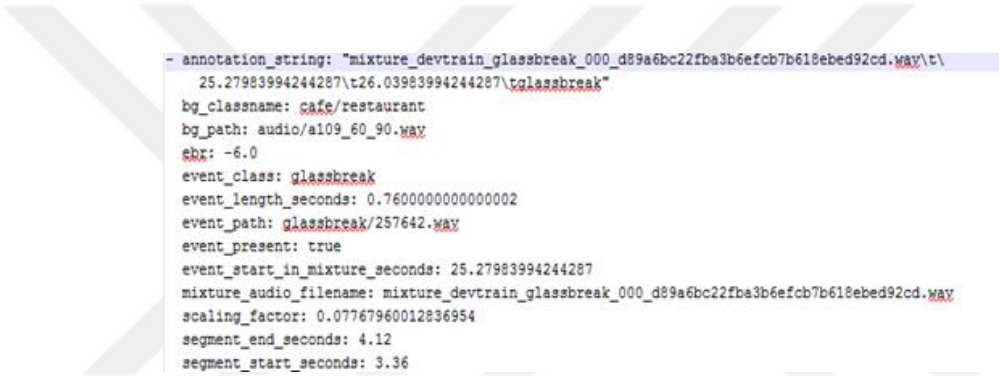


Figure 3.5 DCASE 2017 task-2 development dataset “yaml” file for glass break

Dataset for Environmental Sound Classification (ESC) contains two databases. There are 10 classes and each class has 40 clips in ESC-10 dataset. ESC-10 is subset of ESC-50 which contains 50 classes and each class has 40 clips. There is a readme file in which there is a line for each clip explaining the details of the clip for each dataset [51].

In [52], a dataset of annotated urban sound clips are recorded and taxonomy for urban sound sources is proposed. The dataset contains 10 sound classes with 18.5 hours of annotated sound event occurrences. The dataset contains a CSV file (Figure 3.6) explaining the details of each recording.

```
slice_file_name,fsID,start,end,salience,fold,classID,class
100032-3-0-0.wav,100032,0.0,0.317551,1,5,3,dog_bark
100263-2-0-117.wav,100263,58.5,62.5,1,5,2,children_playing
100263-2-0-121.wav,100263,60.5,64.5,1,5,2,children_playing
100263-2-0-126.wav,100263,63.0,67.0,1,5,2,children_playing
```

Figure 3.6 CSV file explaining Urbansound database

3.3 Database Design

The databases reviewed in Section 3.2 are prepared for just resources of the development of ESR applications. Many of them provide the data and the files for correctly handling the data. In this context they are valuable resources for all ESR researchers. DESRA [45] provides extra tools such as a web graphical user interface for searching and testing.

In the development process of an ESR system, some main functions training, testing and production explained in Section 3.1. During this development process, many sound clips are used; new sound clips may be added or deleted. We deal with many features extracted from these sound clips; we create models using different ML algorithms. Then we compare the models; try to find best features and best ML algorithms. This loop continues. Our first goal in this database design is to help researchers as much as possible to ease the burden of handling data. The second goal is to help maintenance of the data. Our data is sound clips, features, algorithms and models. Addition, deletion, searching, annotation, backup and restore can be thought as the maintenance task.

We should first gather functional and non-functional requirements for the database we will develop. After the requirements are gathered, the database can be developed.

3.3.1 Functional Requirements

The functions for the database are as follows:

- The database should find the impulsive sounds in the recorded audio clips by using the developed algorithms and should record them in the database.
- Impulsive sounds should be embedded into noise clips at the desired SNR in order to make performance comparisons.
- There should be scripts which will extract features of the impulsive sounds.
- These scripts providing the functionality to the database should also be recorded in the database and the functionality should be increased by adding new scripts.

3.3.2 Non-Functional Requirements

Database should contain environmental sounds which consist of impulsive sounds. Every clip should have a type. At the beginning the database should include gunshots, traffic accidents and screams. Regarding each clip, file size, file format, sampling frequency, quantization bit size, recording channel number and origin of the file should be recorded to database. By running an impulse detection algorithm on these clips more than one impulse clips should be found and these impulse clips should be recorded to database. Regarding these impulse clips, it should be known that which sound clip it belongs to, start sample index and start time in that sound clip, algorithm which is used to extract it and the number of samples. There will be no limit on the impulse detection algorithms.

The environmental sounds in which the impulsive sounds will be detected should also be recorded in the database. These sounds can be traffic, crowd, thunder and etc. We will call these sounds environmental noises. Every information requirement written above for impulsive sound clips is required for these too.

There will be sound clips which are constructed by embedding sound clips with impulses into environmental noises described above. Regarding these clips, SNR, the start sample index and the start time of the impulsive sound in the noise should be known.

At first the database contains only three types of environmental sounds namely gunshots, traffic accidents and screams. There will be no limit for increasing it. There will be no limit for entering machine learning and other algorithms to database.

3.3.3 Implementation of the Database

The implemented database will include some data which can be stored by way of data types found in the standard database software and also sound files. These sound files will not be stored to database instead they will be stored in the operating system file structure and the path to this file is just recorded in the database.

Microsoft SQL 2008 Database Server is used to create tables. The tables and the relationships between them are shown in Figure 3.7. The scripts providing functionality are coded in Matlab 2011a. The files are stored in "mat" type when required.

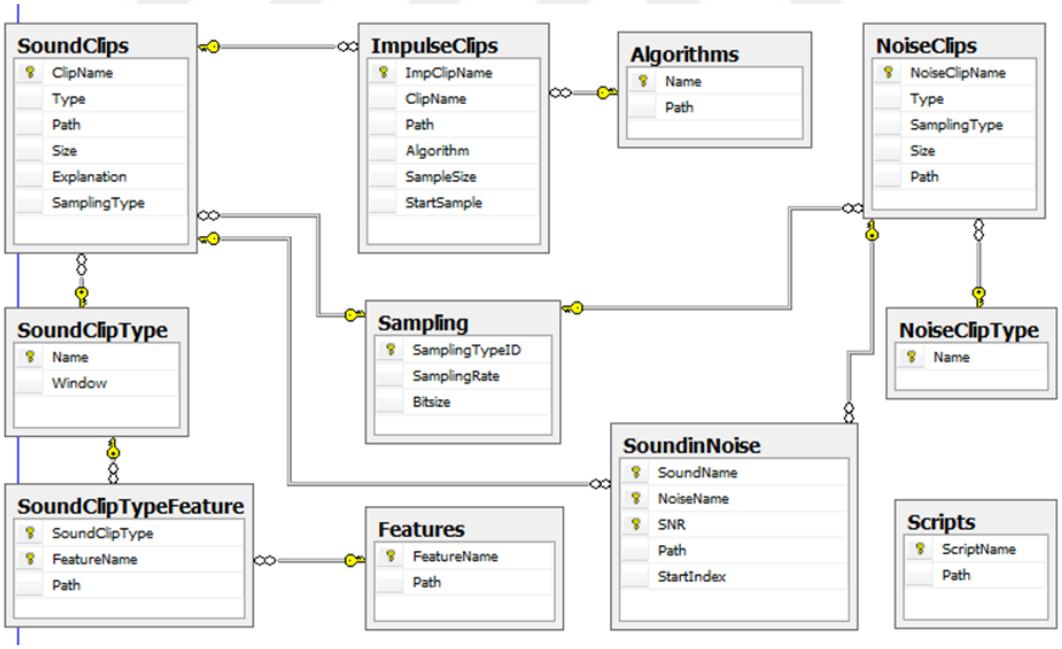


Figure 3.7 Database table diagram

For functional requirements three scripts coded using Matlab:

- *Impulsive Sound Detector:* This script is an interface to algorithms that extracts impulses from the sounds. Different algorithms can be used here for extraction of the impulses. The algorithms must conform to this script interface definition. This script takes the algorithm name that will be started and the environmental sound type in which the impulse will be searched are given as arguments.

- *Impulsive Sound Embedder:* This script takes the type of the impulsive sound, noise type, SNR level and at last the number of required record count to be created. The script merges the environmental sound clip with the noise clip at the desired SNR level and records it to the database table.

- *Feature Extractor:* This script acts as an interface to feature extractors. It takes the path of the feature extractor from the features table, feature name and the impulsive sound type from command line. After extracting features, it is saved as a mat file.

3.4 Database Usage and Tests

To test our database we prepared work flows and we entered data according to these work flows and then we started the related script. We prepared the work flows seen in Figure 3.8. In the first work flow we test the impulsive sound detector script. This script finds the impulsive sound parts of a clip and saves the properties to the related tables. The second script takes the output of the first script namely the impulsive sounds and extract features from these sound and writes the properties of features to the related tables.

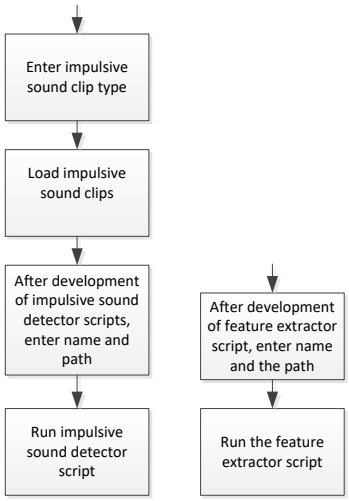


Figure 3.8 Workflows for database testing

3.5 Data for the Database

We found the required sounds from the resources below to continue with the thesis.

- Detection and Classification of Acoustic Scenes and Events (DCASE) 2017 Task 2 database gunshot and glass break sound clips were entered the database. Total 1000 background sounds, 500 gunshots and 500 glass break sounds are loaded.

- Audio Events Data Set for Road Surveillance Applications. Total 200 car crashes, 200 tire-skidding and 5600 scream sounds with traffic background are loaded.

3.6 Database Extension

This database was extended to be able to store the all kind of environmental sounds. A user interface was developed to be accessed by everyone who needs environmental sound clips. A WEB interface was also developed to be accessed from the WEB. [53] Explains the details of this database. We summarize the details of the database in the following sections.

3.6.1 Taxonomy of Environmental Sounds

We need taxonomy to be able to store, search and retrieve the data from the database. During literature review we see two kinds of taxonomy. In the first taxonomy, the sounds can be grouped by the event that makes the sound. In Figure 3.9, grouping of sounds according to this taxonomy defined in http://www.auditorylab.org is seen.

title	author	type	modified
LEGAL NOTICE	Auditory Lab User	Page	2009-06-24 21:16
Impact Events	Auditory Lab User	Folder	2009-06-25 10:15
Deformation Events	Auditory Lab User	Folder	2009-06-25 10:16
Rolling Events	Auditory Lab User	Folder	2009-06-25 10:17
Air Events	Auditory Lab User	Folder	2009-06-25 10:17
Liquid Events	Auditory Lab User	Folder	2009-06-25 10:18
Experiment Stimuli	Auditory Lab User	Folder	2009-06-25 10:19

Figure 3.9 Grouping of sounds based on sound producing events

Another hierarchical grouping is seen in Figure 3.10. This classification method is the same as seen in Figure 3.9. This is the grouping defined in [54] based on sound producing events and the listeners' description of the sound.

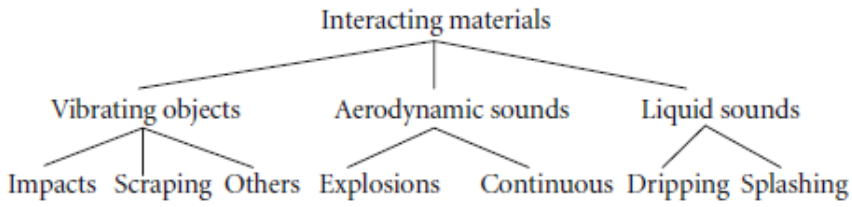


Figure 3.10 Grouping of sounds defined in [54]

Second taxonomy is based on the sound source [52], [55]. The subset of the taxonomy defined in [55] is taken into consideration for urban sounds given in [52] (Figure 3.11).

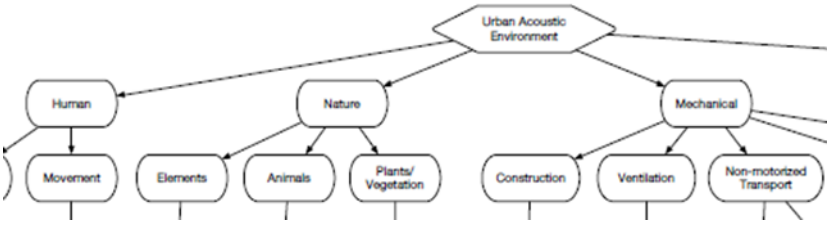


Figure 3.11 A part of the urban sound taxonomy defined in [52]

In our database design we decided to use taxonomy of the sounds like the one described in [52]. In [52] one of the aims is to go to as much detail as possible going down to low level sound sources. In our database design we don't need the steeps between main branch and the leaves. For example instead of human→ movement→footsteps, we use just as human→ footsteps. In Figure 3.12, how the sounds are grouped in our database design is explained.

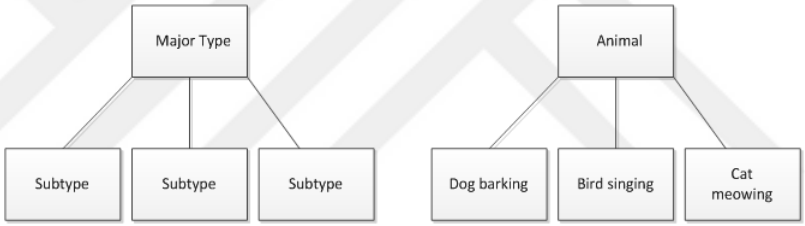


Figure 3.12 Environmental sounds grouping in the database

3.6.2 Non-Functional Requirements

- There will be sound clips in the database. These sound clips will contain environmental sounds. Each sound clip can have more than one environmental sound. These environmental sound clips can belong to different major environmental sound types. Origin of these sound clips, such as own recording, from an internet site or from another database, should be entered to database. Recording details should be entered. Size, file type of sound clip and the path where the sound clips are recorded should be entered to the database.

- There should be major types, such as human, nature, animal, etc. There should be subtypes, such as dog bark, gunshot etc. and these must belong to major type. Start and end sample index or start/end time of these clips should be known.

- There will be environmental sound clips in the database. These clips should be extracted from sound clips defined in a. Each of these sound clips should have a subtype and it should be known from which sound clips it is extracted. Extraction method should be entered. If it is extracted by an algorithm the algorithm name otherwise as “manual” should be entered to database.

- Recording details should be entered to the database such as, sampling rate, quantization bit, channel size, etc.

- Background clips should be entered to the database. Each background clip should have a type. Each background clip should also have file type, file size, recording detail and the path where it is recorded.

3.6.3 Functional Requirements

- There should be scripts which will take algorithms as arguments and extract the environmental sounds from sound clips. These extracted clips should be entered to the database as explained in 3.6.2.

- There should be scripts which will embed environmental sound into noise clips at desired SNR level.

- There should be scripts which will extract features from environmental sound clips and store to the database.

- The scripts, their help files, paths should be stored in the database.

3.6.4 Graphical User Interface (GUI) Requirements

- Administrators can use the GUI for meta data entrance, deletion and update such as major types, subtypes, recording details.

- Administrators can use the GUI for data load manually or using the data loader script.

- Researchers can use the GUI for searching and downloading the desired environmental sounds.

- Researchers can use the database to test their recognition algorithms.

3.6.5 Database Implementation

The implemented database will include some data which can be stored by way of data types found in the standard database software and also sound files. These sound files will not be stored to database instead they will be stored in the operating system file structure and the path to this file is just recorded in the database.

Microsoft SQL 2008 Database Server is used to create tables. The tables and the relationships between them are shown in Figure 3.13. The scripts providing functionality are coded in Matlab 2011a. The files are stored in “mat” type when required.

For functional requirements the following scripts are coded using Matlab.

Environmental Sound Detector: This script is an interface to algorithms that extracts environmental sounds from the sound clips. Different algorithms can be used here for extraction of the environmental sounds. The algorithms must conform to this script interface definition. This script takes the algorithm name that will be started and the environmental sound types which will be searched are given as arguments.

Environmental Sound Embedder: This script takes the type of the environmental sound, noise type, SNR level and at last the number of required record count to be created. The script merges the environmental sound clip with the noise clip at the desired SNR level and records it to the database table.

Feature Extractor: This script acts as an interface between feature extractors. It takes the path of the feature extractor from the features table, feature name and the

environmental sound type from command line. After extracting features, it is saved as a mat file.

Data Loader: To load data from other databases this script is used to interface with data loading scripts.

The GUI is developed using Microsoft Visual Studio with C# language. GUI has admin utilities and end user tools. In Figure 3.14, it is seen the part of GUI providing admin operations. These admin operations are additions and deletions of major types, subtypes and sampling information. Besides the GUI provides the administrators load data one by one or using a script to load as a batch.

The GUI provides the users some tools. The tools are for searching the database and testing their algorithms on the sound clips given by the database. Users can search the database to see the sound clips with desired type.

End users can search the database with noise clip type, sound clip type and with SNR interval then select a clip and copy it to their own computer. After finding the start index of the embedded environment sound clip, by writing the start index of this environmental sound to the edit box on the GUI and by clicking the test button, they can test their algorithm correctness. The WEB GUI for end user tools is seen in Figure 3.15.

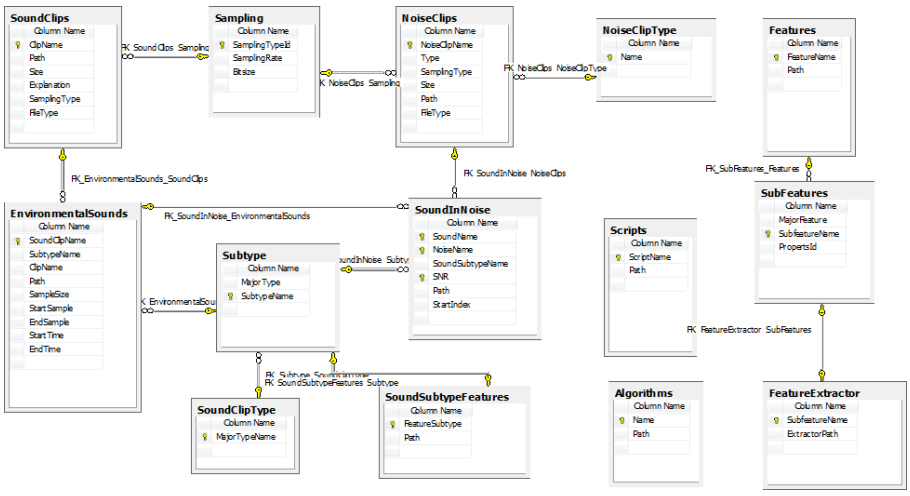


Figure 3.13 Environmental sounds table diagram

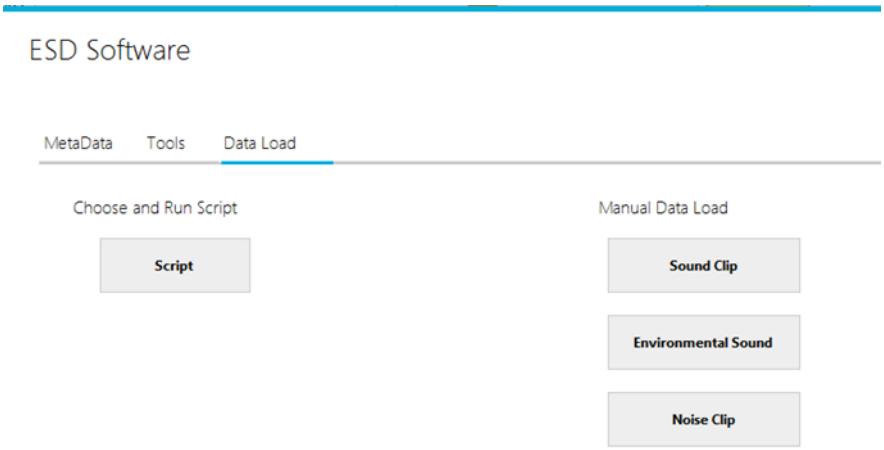


Figure 3.14 Desktop application GUI for administrators to data load

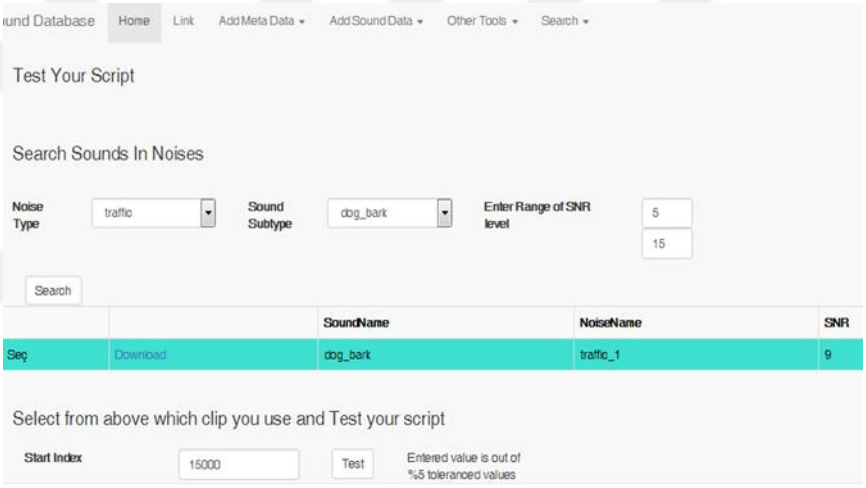


Figure 3.15 WEB GUI for end user tools

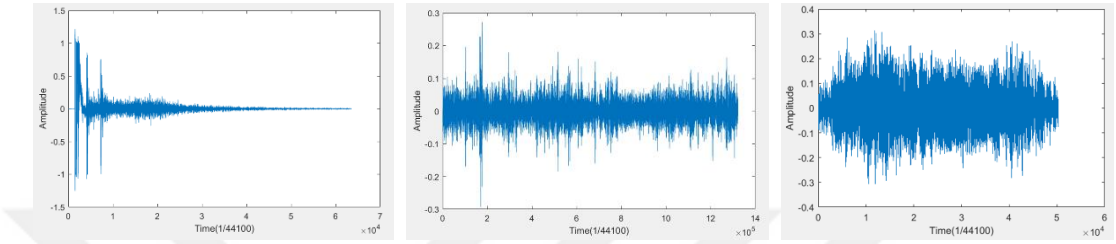
CHAPTER 4

IMPULSIVE SOUND DETECTION

4.1 Introduction

A typical ESR system has the following parts: Microphone input and digitization, detection, feature extraction and recognition. This system can be run in real-time or offline after recording. Generally a surveillance system for hazardous sound event detection works alone or incorporates a video surveillance system. Figure 4.2 is showing a typical ESR system [56]. This system can be used to detect and recognize dangerous audio events. Then, the data flow is as follows. Environmental sounds are captured and digitized by a microphone input and a sound card. Sampling rate and quantization bit depth are adjusted here. Sampling rate can affect the performance, because each sample is handled at the detection stage separately. We should also consider the buffering size here which will affect the performance later. Detection stage comes after the input and digitization of the sound input. Detection stage detects the impulsive sounds. Impulsive sounds are the result of some dangerous events such as gunshots, explosions, human screams etc. or they may be from other sources such as thunder, helicopter [3] (Figure 4.1). In detection stage just framing (windowing) or first framing then sub-framing techniques can be applied. Some algorithms inspect each sample alone to detect if it is an impulse or not without framing. Frame size has a big impact on the detection of impulsive sounds and later in the recognition stage. Detection algorithms always work in the system. If an impulsive sound is detected then the detected frame is passed to recognition stage. For real-time constraints, a dangerous sound event detection algorithm must run near real-time. Another important thing we should consider in detection stage is the threshold value. Threshold is applied on the output of the algorithm for decision. The threshold value should not be very low to overload recognition stage or it should not be very high to miss dangerous sound events. The main objective of the detection stage is to decide on frames which consist an impulsive sound and then to pass these

frames to the recognition stage. On the other hand in some systems detection stage can be omitted. In the next section we will see some algorithms which we classify as detection by recognition. ESR systems using these algorithms don't need the detection stage [57],[58]. After detection of impulsive sounds, recognition stage takes the frame and decides if it is from a dangerous event or not and the class of the event.



(a) (b) (c)

Figure 4.1 Graphs of (a)gunshot (b)traffic accident (c)scream

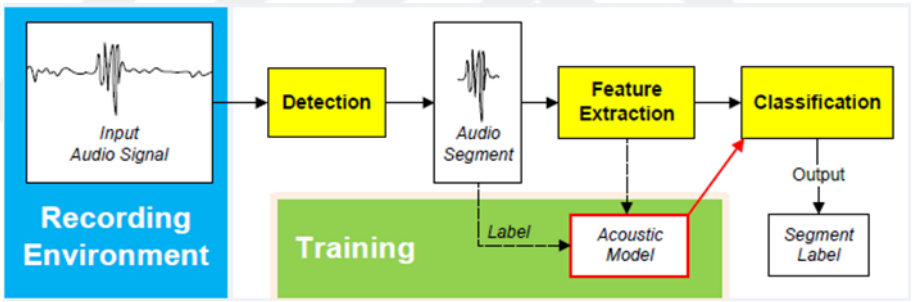


Figure 4.2 A typical ESR system [56]

In the next section we will explore the algorithms used at the detection stage. We will also study on the usability of impulsive noise detection algorithms at this stage.

4.2 Classification of Impulsive Sound/Noise Detection Algorithms

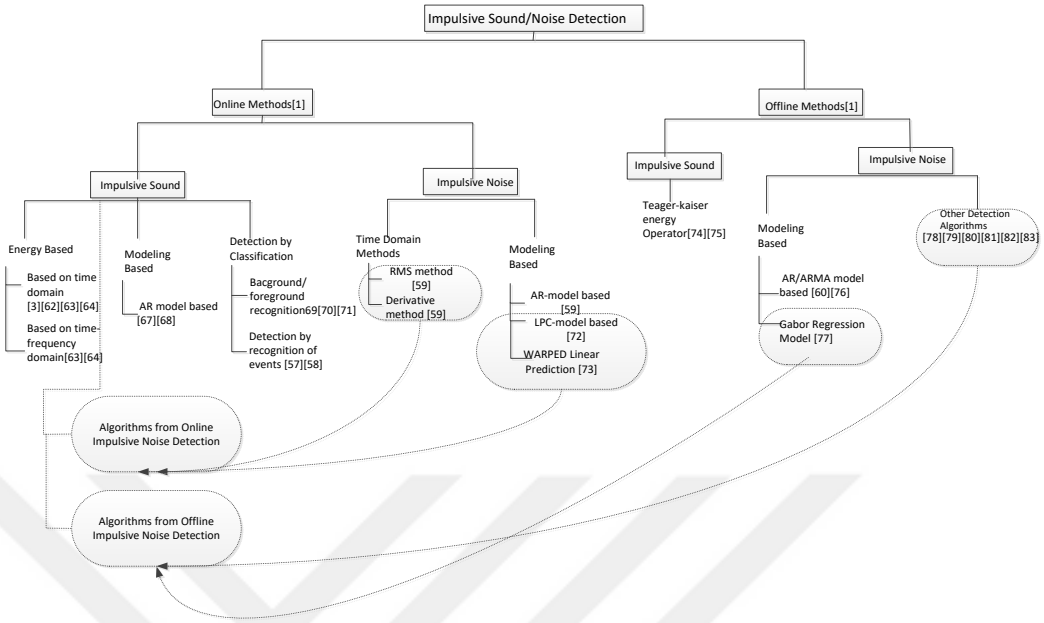


Figure 4.3 Taxonomy of Impulsive Sound/Noise Detection Algorithms and Our New Approach for Impulsive Sound Detection

In Figure 4.3 a classification of impulsive sound/noise detection algorithms is shown. Impulsive sound/noise detection algorithms can be grouped into two at the top, regarding the detection time. Online or real-time algorithms detect the impulsive sound/noise during the dangerous event happens. Offline algorithms detect it from the sound recordings of the event after the event has happened [56].

Furthermore the detection algorithms can be divided into two subclasses, as impulsive sound and impulsive noise detection algorithms. Impulsive sounds are the results of an acoustic event, they possess high energy and they last longer than impulsive noises. They have certain patterns so they can be classified by machine learning algorithms. Impulsive noise occurs in analog transmission media such as communications, telephony, and radio broadcast caused by electromagnetic disturbances and atmospheric noise. Impulsive noise results also from physical damages to the storage media e.g. degraded gramophone recordings [59]. Impulsive

disturbances or clicks can be described as short time (1 to 200 samples at 44100Hz sampling rate) local discontinuities [60].

Hazardous events such as gunshots, explosions, glass breaks, door slams generate impulsive sounds. Audio surveillance applications which are based on impulsive sound detection and the applications which enhance the sound quality of recording media and communication try to detect impulsive sound and impulsive noise respectively using their own algorithms.

It should be noted here that there is no restriction to use online impulsive sound or noise detection algorithms for offline applications.

4.3 Online Impulsive Sound Detection Algorithms

4.3.1 Algorithms based on signal energy

The most common parameter that can be used to discriminate an impulsive sound from the background is its energy. The sounds that we are interested in such as gunshots, door slams, screams cause an abrupt change in audio volume depending on the typical auditory scenarios. Following this principle, some works proposed to segment the audio stream into fixed-length windows and discard all the windows whose energy is below a predefined threshold [61]. In [3], it is proposed three methods using above principle. A real-time environmental sound recognition system is implemented in [7] for Android operating system. In this system an energy detector which is based on one of the algorithms defined in [3] is used and reported that it was fast and efficient. For robust localization of impulsive sounds, an energy detector is used first to detect impulsive and non-impulsive sound sources in [62]. Besides these algorithms which run in time domain, there are other types of algorithms using this principle running in time-frequency domain [63], [64].

Energy based algorithms are generally fast compared to the others but there are problems related to these algorithms, described below:

Threshold adjusting: Actually this problem is related to most of the detection algorithms. Impulsive sound decision is given based on a threshold value. If the energy of current window is above the threshold value then the window contains impulsive sound. The threshold value can be a fixed value or adaptive. This value should be set correctly so that the system would not miss impulsive sounds. We call this miss detection. If the threshold level is too high then miss detections occur. On the contrary if the threshold level is too low then we detect other sounds we are not interested in. It is called false positives. This puts the recognition stage an extra load and the system slows down. To determine the threshold value, fixed threshold value [65], set of thresholds [66] and adaptive thresholding [3] was proposed.

Windowing: The algorithms described above based on energy, segments the signal into fixed length blocks called windowing. This fact may chop a significant audio event into two adjacent blocks so making it more difficult for the subsequent processing stages [61]. For example at the recognition stage features must be extracted from the complete sound of the acoustic event for correct recognition.

The other problem regarding windowing is that every impulsive sound has different length. So, selecting one window size doesn't match for all the impulsive sounds.

4.2.2 Algorithms based on sound modeling

In [67] and [68], to detect impulsive sounds from the muzzle blasts of gunshots, Auto Regressive (AR) model is used. Although AR model operates on time domain data, it is capable of encapsulating both spectral and energy characteristics of time series. AR model doesn't need any segmentation of the signal. To accurately model the background signal, Non-Stationary AR model is proposed as the most appropriate model. Least Mean Square (LMS) is used to find the parameters of the Non-Stationary AR model [67] and [68].

4.2.3 Detection by classification

We can divide the methods in this category into two. The first category combines the methods that classify the sounds in the environment as background or foreground,

later the algorithm tries to recognize foreground sounds which are called abnormal sound events. In this first category, it is generally assumed that there is no knowledge of foreground (abnormal) sounds hence it is not possible to train the algorithm in advance. The algorithm learns the background generally online, and then it can detect the abnormal events in the environment. In [69], a Gaussian Mixture Model (GMM) is proposed for modelling the background sounds. This model updates itself incrementally when the new sounds come. The proposed model is a hybrid one which detects the foreground suspicious activity online and then this suspicious activity is classified after inspection as usual or unusual activity.

In [70], the Time-Adaptive mixture of Gaussians method, well-known in the video surveillance context, which aims at discovering the deviance of a signal from the expected behavior in an online fashion, is used. Windowed audio signal is transformed to frequency domain and the energy of different sub bands are modeled by different GMMs. This model is also adaptive to background sounds and can detect abnormal sounds.

In [71], a technique that uses multiple GMMs for different level of sound events is explained. The basic idea of the multi-stage GMM is to model the majority of the training samples using the first GMM, and then model the rest of the samples using the second GMM. The training stage goes until usual sounds are classified by the end of the GMMs. If a sound signal comes with likelihood less than the last GMM's, then this sound is classified as abnormal.

The methods in the second category of detection by classification can classify the sounds into predefined and trained classes. For real-time applications the computation time must be considered for this category. In [57], it is proposed that the observed sound signal is composed of signal and noise and the signal and noise subspaces are orthogonal to each other. As a consequence of this, noise effect is reduced by projecting the observation vector onto signal subspace. During the training phase Principle Component Analysis (PCA) technique is used to create event subspaces. To construct feature vectors Short Time Fourier Transform (STFT) is

taken and PCA is calculated by including 90% of the signal energy. These features are used with Support Vector Machine (SVM), GMM and Deep Belief Network (DBN) to recognize events such as scream, glass break, and siren. It is shown that the proposed features outperform conventional features such as MFCC when used with the same algorithm.

In [58], one-class SVMs are proposed for audio surveillance systems. This paper explains a feature vector selection algorithm and constructing a fixed length feature vector from different duration signals. Lastly it is explained that wavelets are appropriate as features for short duration and non-stationary signals.

4.4 Online Impulsive Noise Detection Algorithms

4.4.1 Time domain methods

There are two algorithms in this category. Derivative method uses the definition of impulsive sound which is sudden change between successive samples. So, derivative method is based on tracking this change. Fourth derivative gives more accurate results to detect the impulsive noises. Also in [59] an adaptive threshold is applied on the pre-determined number of past derivative values.

Other method is the Root Mean Square (RMS). The root mean square of the windowed samples is calculated [59].

4.4.2 Sound modeling based

In [59], previously explained AR method is used first to predict the incoming samples. AR order is taken as 10. The AR prediction errors are found and then to select the impulsive samples previously explained derivative method is used. The difference between this algorithm and the previous is that while this algorithm accepts the signal as stationary, the other accepts non-stationary. This difference leads to calculation of model parameters differently.

In [72], it is declared that the Linear Predictive Coding (LPC) is a good way to model speech signals whereas it is not good for impulsive disturbances. LPC order is taken as 20. After detecting impulsive noises, an interpolation algorithm is proposed to replace noisy samples. It is reported that it gives successful results on numerous noisy recordings.

It should be noted that there is little difference between AR model and LPC model. AR and LPC model coefficients are the same if you take the same order. But, the interpretation is different.

In [73], how Warped Linear Prediction (WLP) is used to detect impulsive noises. WLP operates in the frequency domain and to detect the impulsive noises changes the prediction gain. A comparison is made in [73] between LPC and WLP and it is shown that WPC decreases miss and false detections when the parameters are chosen correctly.

4.5 Offline Impulsive Sound Detection Algorithms

Offline impulsive sound detection may be useless, because the aim of detection is to respond to real time events immediately. So there are just a few researches on offline impulsive sound detection. These impulsive sounds are not from dangerous audio events.

In [74], Teager-Kaiser (TK) operator is used to detect the clicks of sperm whales in large recordings. Selection of clicks is based on applying a statistical measurement on TK operator output. As a result 94.5% correct detection was achieved.

In [75], to distinguish prerecorded sounds of six species of odontocetes TK operator was used. Sound recordings are windowed and converted to frequency domain, and then TK operator is applied on the high frequency components. Detected windows are recognized using GMM.

4.6 Offline Impulsive Noise Detection Algorithms

4.6.1. Algorithms based on sound modeling

As in online impulsive sound detection, AR algorithm is used to model the background sound signal. In [76], AR method is used to detect impulsive noises such as clicks, bursts or scratches. The method is tested on manually distorted several types of sounds (classical, jazz, vocal etc.) and shows good performance.

In [60], detailed theoretical background of AR/ARMA model is given and explains how to remove clicks from sound signals. AR model can be extended with ARMA given more accurate localization of click.

In [77], Gabor regression model is used to determine the positions of impulsive noises. Just as Fourier Transform is a representation for stationary signals, Gabor representations are appropriate for slowly time-varying signals [77].

4.6.2. Other detection algorithms

There are many different impulsive noise detection algorithms in literature that can't be grouped into small number of groups. Here we mention just the most recent ones. In [78], five different outlier detection algorithms and an outlier detection algorithm which utilizes prediction algorithms for univariate time series are proposed. Proposed outlier detection algorithm is tested with twelve different forecasting models. The proposed Absolute Predictive Deviation (APD) gives the best result when used with the Autoregressive Integrated Moving Average (ARIMA) model.

In [79], a bidirectional processing is proposed. In this new approach, noise pulses are localized by combining the results of forward-time and backward-time signal analysis. For forward-time and backward-time signal analysis AR model is used.

In [80], the proposed semi-causal and noncausal outlier detection techniques are demonstrated on stereo gramophone recordings.

In [81], detection of noise pulses in stereo recording is achieved by using vector AR model and Kalman Filtering.

In [82], it is shown that semi-causal/non-causal solutions based on joint evaluation of signal prediction errors and leave-one-out signal interpolation errors improve detection results compared to prediction-only based solutions.

In [83], the method called binary time-frequency masking threshold criterion is proposed to detect and remove Gaussian, Supergaussian and impulsive disturbances. The noisy signal is decomposed into sum of sinusoids and a residual. AR model is used on the residual to detect the impulsive noise.

4.7 Impulsive Sound Detection Methods

By using above algorithms to construct a surveillance system for hazardous sound events, the detection stage possibilities are explained here.

The first method is seen in Figure 4.4.

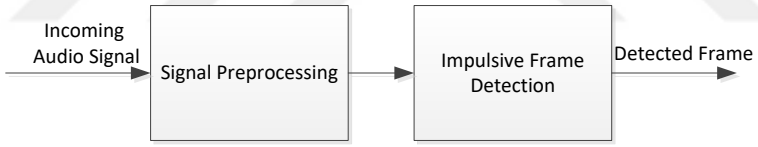


Figure 4.4 Hazardous Sound Event Detection Method 1

In Figure 4.4 impulsive sound detection block can use one of the algorithms from;

- Energy based algorithms, [3], [62], [63], [64].
- Prediction based algorithms, [59], [60], [67], [68], [72], [73], [77].
- Other impulsive noise detection algorithms, [78], [79], [80], [81], [82], [83].

Second method is based on foreground-background subtraction as seen in Figure 4.5.



Figure 4.5 Hazardous Sound Event Detection Method 2

In this method after preprocessing, a machine learning algorithm can be used to differentiate between background and foreground sounds. The foreground sounds for our application are sounds of interested hazardous events. The application can be trained in advanced or online for required impulsive events. In this method the algorithms defined in papers [69], [70] and [71] can be used.

Third method is shown in Figure 4.6.

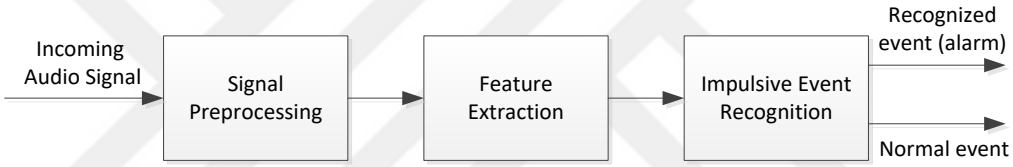


Figure 4.6 Hazardous Sound Event Detection Method 3

In this method event recognition block is trained before for required events. It recognizes the events by inspecting features extracted from incoming signal. The algorithms defined in papers [57] and [58] can be used here.

4.8 Implementation of Impulsive Sound/Noise Detection Algorithms

We implemented two energy based algorithms, three prediction based algorithms and an algorithm using wavelet.

4.8.1 Energy Based Algorithms

Detection using Normalized Power Sequences:

The algorithm defined in [3] is mainly implemented, but some parameters are changed. The process is as follows:

1- Incoming samples are framed. (N = 128) (Figure 4.7)

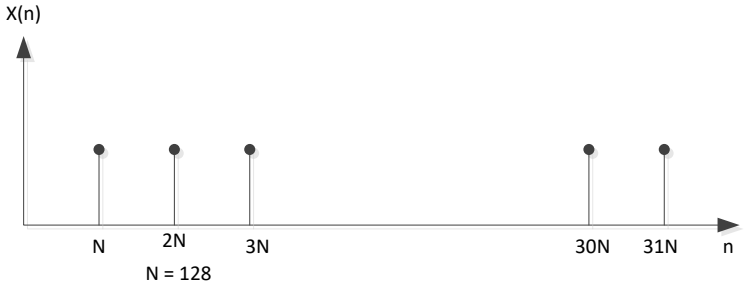


Figure 4.7 Framing of incoming samples

2- Power is calculated for each coming frame. (Figure 4.8)

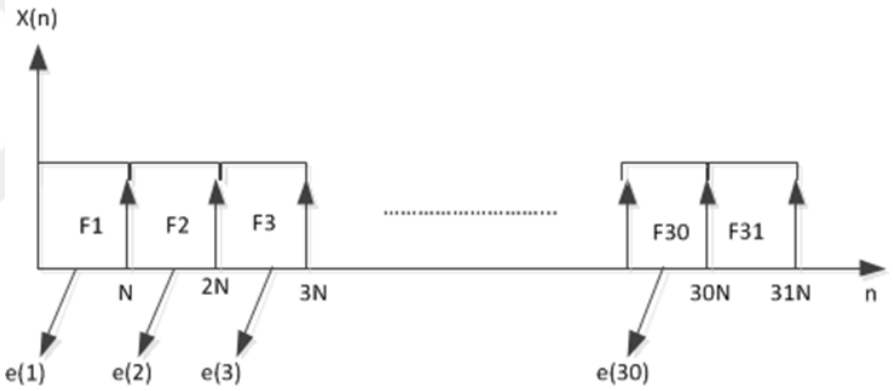


Figure 4.8 Power calculation

$$e(k) = \frac{1}{N} \sum_{n=0}^{N-1} x^2(n + kn) \qquad k = 0,1,..... \qquad (4.1)$$

3- Powers are windowed and normalized. (Figure 4.9)

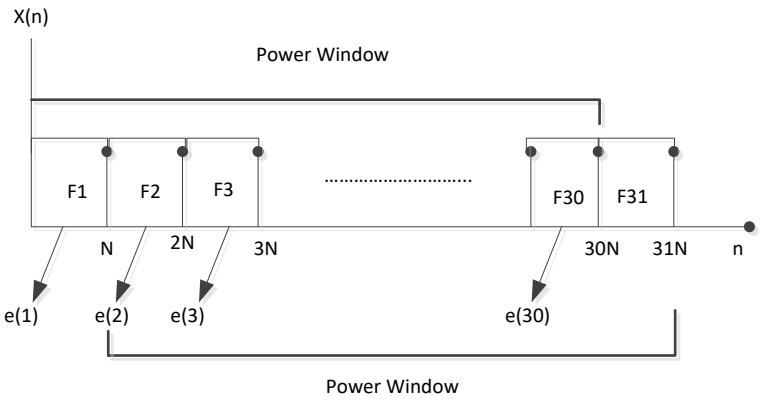


Figure 4.9 Powers are windowed and normilized window size = 30

$$e_{norm}(j) = \frac{e_{win}(j) - min_j(e_{win}(j))}{max_j(e_{win}(j) - min_j(e_{win}(j)))} \qquad J= 0,1, ..., L-1 \quad L = 30 \qquad (4.2)$$

4- Calculate standard deviation of normalized power window.
   {F1, F2, …, FN}→{e(1), e(2), …, e(N)}→{enorm(1), enorm(2), …, enorm(N)}→σ{Enorm}
5- If σ{Enorm} < 0.3 then F_N contains impulse.
6- Slide window j to the right.

Detection Using a Threshold on the Power Sequence:

The algorithm defined in [3] is mainly implemented, but some parameters are changed.

1- Incoming samples are framed. (N = 128)
2- Power is calculated for each coming frame.
3- Powers are windowed i= {e(1), e(2), …, e(N)} N= 30;
4- Threshold is calculated using power window as follows:

$$th(i) = k1*std(i) + k2*mean(i)$$

5- If power of frame (N+1) > th(i) then frame (N+1) contains impulse.
6- Slide window i to the right.

4.8.2 Prediction Based Algorithms

Linear Predictive Coding (LPC):

LPC implementation first finds the LPC coefficients by using a prediction order 5. These coefficients can be used for a period of the sound which is assumed to be stationary. Then calculation is redone. Filtering the signal using LPC coefficients an LPC estimate of the signal is found. Difference between the original and the estimate gives the error. By applying a threshold on to the error, impulsive noise is obtained. LPC formulation steps are given as follows:

1- A sound signal modeling by LPC is:

$$x_n[\mathrm{n}] = \sum_{i=1}^{N} a_i x_i[\mathrm{n} - \mathrm{i}] + e_n \qquad (4.3)$$

2- If there is an impulsive sound:

$$y_n = x_n + e_n + d_n \qquad (4.4)$$

3- By passing y_n from a Finite Impulse Response (FIR) filter we can obtain error with impulsive sound. FIR transfer function is:

$$H(z) = 1 + \sum_{i=1}^{N} \bar{a}_i z^{-i} \qquad (4.5)$$

4- We can detect the impulsive sound if the error exceeds a threshold value.

The details of LPC can be found in [84].

Warped Linear Prediction (WLP):

In WLP the delay elements of the FIR filter in LPC are replaced by first order all-pass filters [73]. Then the delay element is as follows:

$$\tilde{z} = \frac{z^{-1} - \lambda}{1 - \lambda z^{-1}} \qquad |\lambda| < 1 \qquad (4.6)$$

By replacing in this new formula just the delay element with λ = -0.7 and applying the same formulation in LPC, impulsive sounds can be obtained. By applying the same parameters as in LPC, it is seen that WLP increases true positives and decreases false positives, as described in [73]. We implemented the WLP using the tools given at [85].

Gabor Regression:

Gabor expansion or transform of a time series is given as follows:

$$y_t = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} c_{m,n} \tilde{g}_{m,n}(t) + d_t \qquad (4.7)$$

Here $\tilde{g}_{m,n}(t)$ are the Gabor synthesis atoms, M is number of modulations and N is number of time shifts, d_t is the noise term. The term $c_{m,n} = < x, g_{m,n} >$ and $g_{m,n}$ is the dual of $\tilde{g}_{m,n}$. In [77], the Bayesian estimation method for Gabor coefficients ($c_{m,n}$) explained in [86] and the method for interpolation of missing data in audio signals explained in [87] are extended to detect the impulsive noises in audio signals. In [77], it is assumed that d_t contains both Gaussian noise and impulsive noise.

We implemented Gabor regression by using the tools given at [88]. We did not use the Bayesian estimation for finding the impulsive noise but we used a method similar to one described in [89]. The method in [89] uses time-frequency filtering which is done by modifying the Gabor coefficients before reconstructing the signal. In our case, by selecting M and N appropriately we found Gabor coefficients and then reconstruct the signal without changing the coefficients. This operation does not produce the exact signal but leaving a residual. We applied a threshold on this residual to find the impulsive sound.

4.8.3 Detection Using Discrete Wavelet Transform (DWT)

The algorithm defined in [63] is mainly implemented, but some parameters are changed.

1- Incoming samples are framed. (N = 512)

2- DWT coefficients are calculated for each frame. Following parameters are applied:
   Daubechies wavelets with 6 vanishing moments,
   Level 6, detailed coefficients are used.
3- Power is calculated for each frame DWT coefficients.
4- Powers are windowed i= {e(1), e(2), …, e(N)} N= 10;
5- Threshold is calculated using power window as follows:

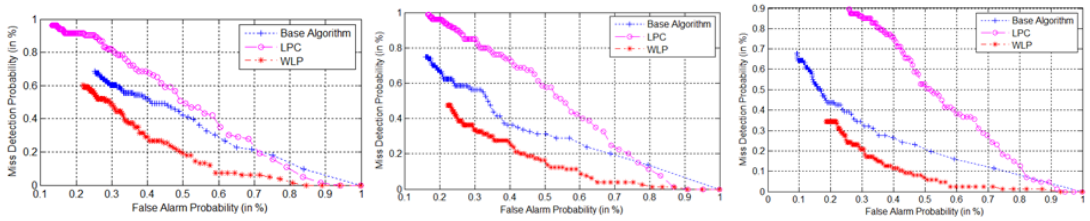
   th(i) = k1*std(i) + k2*mean(i);

6- If power of frame (N+1) > th(i) then frame (N+1) contains impulse.
7- Slide window i to the right.

4.9 Impulse Detection Algorithm Performances

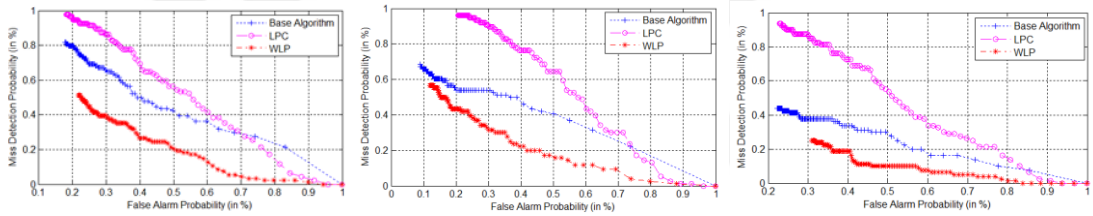
This is a preliminary implementation and the aim is just to see if impulsive noise detection algorithms can be used for online impulsive sound detection. For this purpose we prepared a test sound database from Detection and Classification of Acoustic Scenes and Events (DCASE) 2017 Task-2 database. Task-2 database contains rare sound events which are gunshot, baby cry and glass break. For the tests we used gunshot and glass break sounds. For detection task, these sounds are mixed with 15 different background sounds. Totally there are 250 mixed sounds with a length of 30 sec from each sound class. Each mixture is prepared at one of 3 different event-to-background-ratios (EBR) levels. These levels are -6 0 and 6. The EBR was defined as ratio of average RMS Error (RMSE) values calculated over the duration of the event and the corresponding background segment on which the event will be mixed respectively. The detailed explanation of the dataset can be found in [90]. These sound clips are loaded to the database developed in [1] for easy usage. The algorithm implemented in [3] that is threshold on the power sequence is used as the base algorithm for our tests, because we saw in previous tests that this algorithm was performing well to detect impulsive sounds. Comparison is made based on miss detections (MD), and false positives (FP). MDs are the embedded gunshot sounds which can’t be detected. Correlate of this is true positives, which means correctly

detected embedded gunshot sounds. FPs are possible impulses which are not embedded by us. It should be noted here that these FPs can actually be impulsive sounds but it is not important for us and it does not show that algorithm is running incorrectly. FP should be as low as possible.



(a) (b) (c)

Figure 4.10 DET curves of energy based algorithm (base algorithm), LPC and WLP for detection of gunshot for different EBR values. (a) EBR = -6, (b) EBR = 0, (c) EBR = 6



(a) (b) (c)

Figure 4.11 DET curves of energy based algorithm (base algorithm), LPC and WLP for detection of glass break for different EBR values. (a) EBR = -6, (b) EBR = 0, (c) EBR = 6

Detection tasks can be viewed as involving a tradeoff between FP and MD. Detection Error Tradeoff (DET) [91] curves show the performance of an algorithm by showing the MD and FP rates on the same graph for a set of different threshold values. As the algorithm DET curve closes to the origin performance increases, otherwise it decreases.

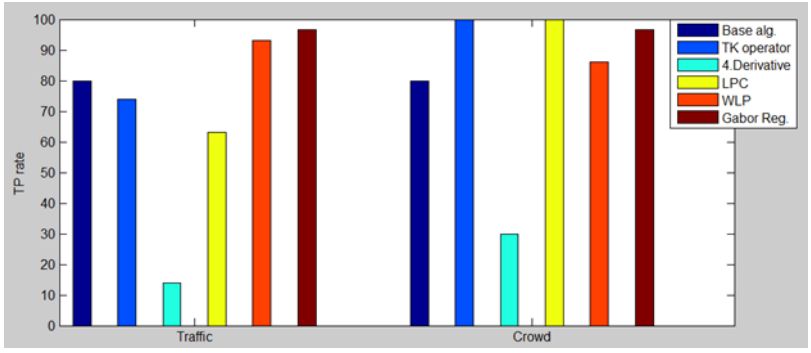


Figure 4.12 TP rate of algorithms for traffic and crowd noise type

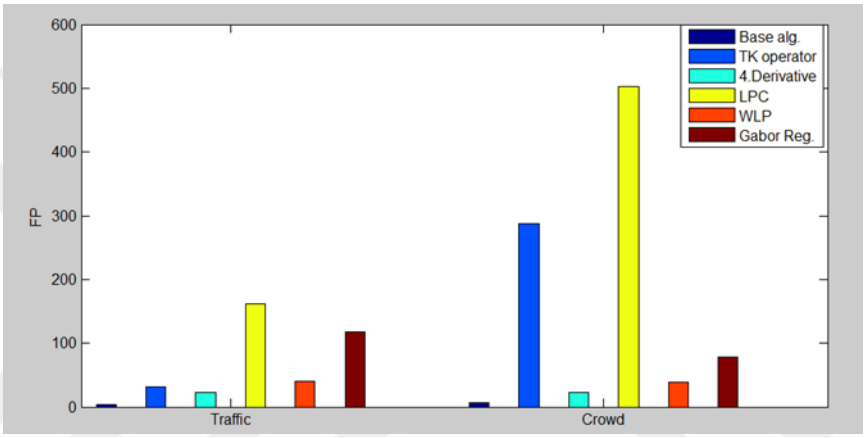


Figure 4.13 FP of algorithms for traffic and crowd noise type

4.10 Discussion

In this section, some algorithms which are not used for impulsive sound detection before, are implemented and tested. Performances of these algorithms are compared with an energy based algorithm. The results are shown in Figure 4.10, Figure 4.11, Figure 4.12 and Figure 4.13. It is shown that base algorithm stands between LPC and WLP algorithms. The WLP algorithm outperforms energy based algorithm (base algorithm).

The base algorithm employs windowing. Windowing has several disadvantages. One is the impulsive sound may chop next window, so it may be difficult to determine start and end of the impulsive sound. The second is even if it is in the detected

window the start of impulsive sound is not exact, somewhere in the window. This ambiguity in start of impulsive sound will affect the performance of recognition stage and location finder stage if employed. More importantly the window length predetermined for a single dangerous audio event such as gunshot may not be appropriate for other impulsive sound events. The detection performance achieved by a windowing length determined for a gunshot may not be achieved for explosions.

All algorithms classified under impulsive noise detection (Figure 4.3) work without windowing. They directly detect the start of impulsive sounds. So they don’t possess the disadvantages of windowing.

CHAPTER 5

IMPULSIVE SOUND RECOGNITION

5.1 Machine Learning

The algorithms that will be employed for recognition of gunshot, car crash and scream will be machine learning algorithms. Here we will explain what machine learning is and how the algorithms are running.

Machine learning systems are made up of three major parts, which are [92]:

Model: The system that makes predictions or identifications.

Parameters: The signals or factors used by the model to form its decisions.

Machine learning algorithm: The system that adjusts the parameters by looking at differences in predictions versus actual outcome.

The machine learning model is created by training. During training, examples or samples are taken from the database, features are extracted from these examples, and the model uses these features to produce the estimated outputs. Then these estimated outputs are compared with the ground truth and the parameters used by the model are updated. At last the parameters given the best results are found and the training of the machine learning algorithm is finished (Figure 5.1).

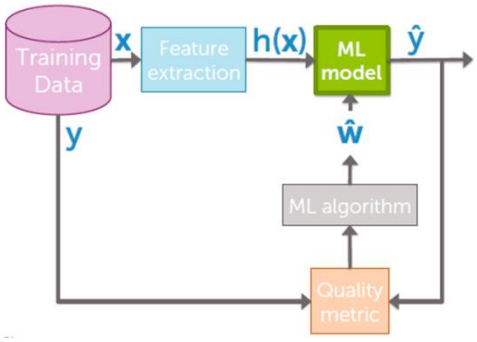


Figure 5.1 Machine learning training

The developed model during the training stage is tested at the testing stage. During testing, again samples or examples are taken from the databese which are not used in training and the performance of the model is found. Figure 5.2 shows the testing stage.

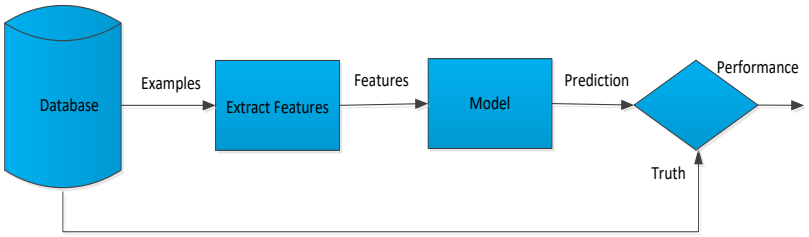


Figure 5.2 Machine learning testing

If performance is good then model can be used for the real purposes, if performance is low then some parameter optimization methods can be applied. Machine learning algorithms can encounter bias or overfitting which will degrade the performance.

After finding the best parameters, machine learning model can be employed to real world input data. In this case we do not know the ground truth and we rely on system outputs. For the system we are trying to develop the machine learning block diagram is seen in Figure 5.3.

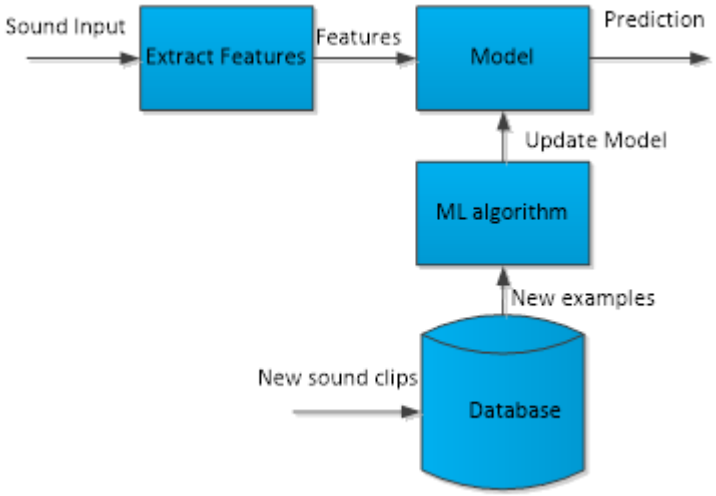


Figure 5.3 Machine learning in usage

5.2 Cost Function

The parameters of the model are learnt from a cost function. The cost function is based on the difference of ground truth and model output. One of the cost functions is:

$$J(\theta_0 + \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^i) - y^i)^2 \quad (5.1)$$

The goal of the machine learning algorithm is to minimize the cost such that:

$$\underset{\theta_0,\theta_1}{\text{minimize}}\ J(\theta_0, \theta_1) \quad (5.2)$$

5.3 Gradient Descent Algorithm

Gradient Descent Algorithm is one of the algorithms which are used to minimize the cost function. It works as follows:

repeat until convergence {

$$\theta_j := \theta_j - \propto \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

for j = 1 and j = 0

}

5.4 Deep Neural Networks

Deep Neural Networks (DNN) is the most popular algorithm in recent years. DNN is a Neural Network with more than one hidden layer as shown in Figure 5.4.

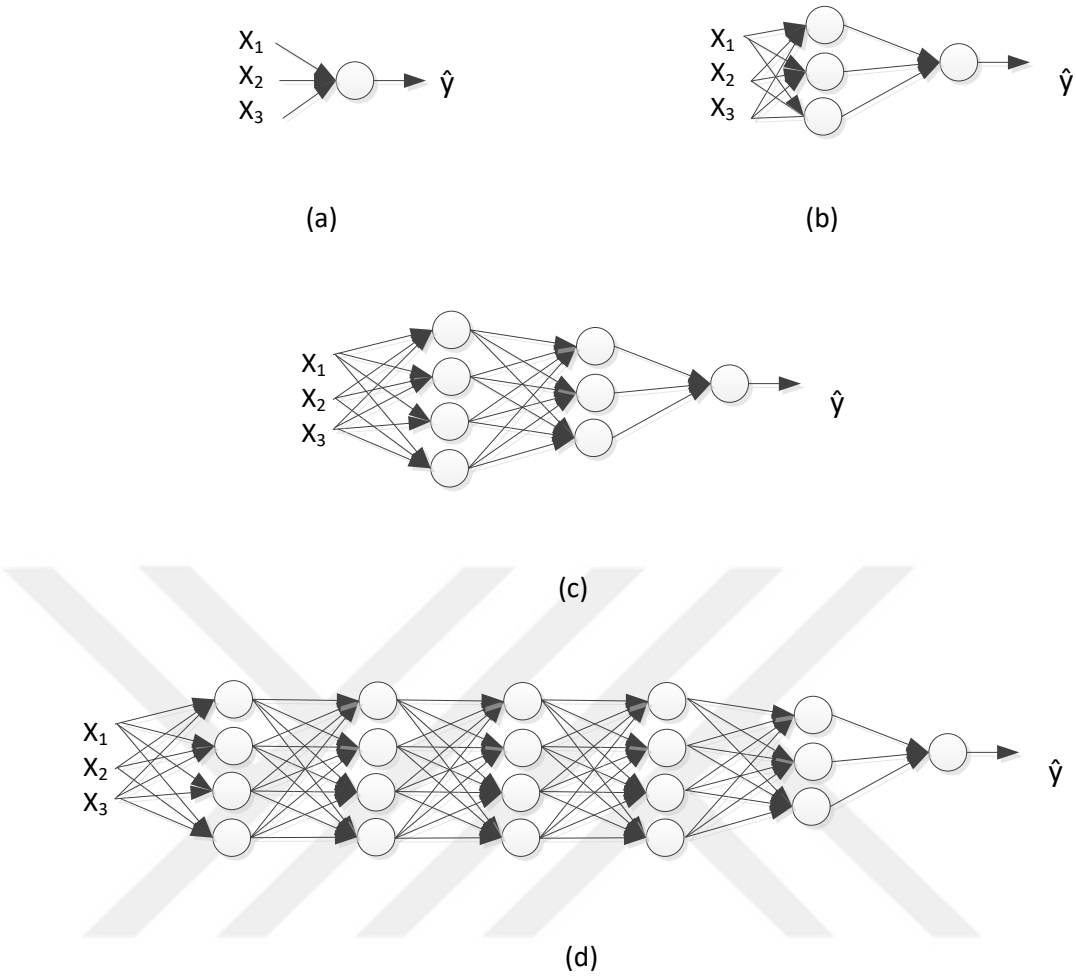


Figure 5.4 (a) Logistic regression (b) 1 Hidden layer NN (c) 2 Hidden layers (d) 5 Hidden layer-Deep NN

DNN learning comprises forward propagation and backward propagation stages. During forward propagation DNN calculates the prediction and the cost and during backward propagation it calculates the gradient of the parameters.

It is called activation function which takes an input and produces an output for the next layer as shown Figure 5.5.

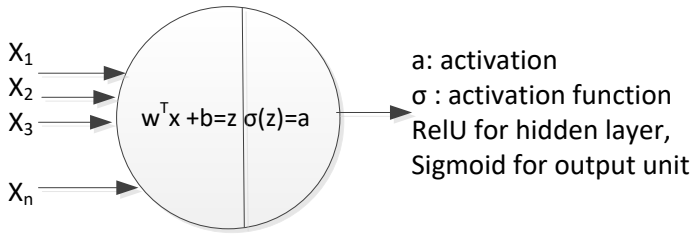


Figure 5.5 A DNN node and the calculation in it

5.5 Implementation of DNN for This Thesis

The DNN is implemented in Matlab 2011a by porting the Python codes from Andrew NG Coursera course. After completing the assignments given in the course for Python and after testing the code developed for the assignments, all code from Python converted to Matlab code.

5.6 Recognition of Gunshot, Scream and Car Crash with DNN

We used DNNs to detect gunshot, screams and car crashes. Audio surveillance applications for security are expected to detect these sounds. There have been many researches on detection of these sounds on offline (recorded) data but online detection of these sounds is important. Our research has the contribution that we are making detection on recorded data and online detection on the same data and comparing the results. During online detection we realized some important points which will help practitioners who want to implement hazardous SED for surveillance applications. Our tests show that the problems encountered in distant speaker recognition (DSR) are also seen in hazardous SED. Current research on SED usually just takes into account background noise during model development. Our online tests show that reverberation can degrade performance significantly. As the results of online tests, we give some recommendations for a hazardous SED development model for the usage in real world applications.

We first try to develop algorithms for an SED system for surveillance. The difference of surveillance systems from the others is that they should respond in near real-time

and they should actually run not on recorded but online data. The other studies done before for the acoustic surveillance worked on the recorded data. These studies take into account just the background noise as the performance criteria. The event sounds are embedded into different background noises such as traffic, metro station, park etc. at different SNR levels. During online testing we see that another factor which is reverberation is affecting the performance. Reverberation is a known factor which degrades the performance; here we show it's degradation by implementation. While the hazardous SED will be used eventually for real-time applications, we propose the reverberation effect to be considered during training and testing.

The SED algorithms for surveillance should aim to work in real environments. After the algorithm has been developed and tested on offline data, the algorithm should be evaluated in online use case where the system capturing audio in real-time. The same pre-recorded data used in offline case can be played back in laboratory environment to verify that the SED algorithms perform similarly as in offline case [93]. For this task we perform two experiments. First experiment tries to find gunshot, scream and car crash in the recorded data. Second we play this recorded data, use a microphone for capturing sound input and the same algorithms are used to detect the interested sounds.

5.6.1 Data, Feature Extraction and Model Preparation

SED recognition part consists of two stages. First stage is to represent the sound which is done by feature extraction and the second stage is recognition. The representation of sound is done by taking fixed length frames of the sound and then by extracting features from these frames. The formal definition of this process can be done as follows: Let's say X is the matrix of acoustic features extracted from the sound signal and $X \in R^{KxL}$ where $K \in N$ is the number of features in each frame and $L \in N$ is the total number of frames in the sound signal. The recognition task is to find probabilities of the frame, $p(y \mid X, \theta)$ for target output vector $y \in R^{L}$ and θ is the parameters of the model. Then by applying a threshold on the frame probabilities, we

can find the estimated event ŷ, if ŷ = 1 the event is present in the frame or if ŷ = 0 the event is not present.

To begin classification, parameters of the model have to be found. This is done through training of the model by giving the labeled features extracted from the frames of the training data. This is called supervised learning. The data we used for training and testing purposes are obtained as follows.

Gunshot dataset is taken from DCASE 2017 challenge Task-2 dataset [90]. We have taken the background sounds, isolated gunshot sounds and mixed sounds which consist background and isolated sounds in 3 EBR values. These levels are -6, 0 and 6. These sound clips are loaded to the database developed in [1] for easy usage. Table 5.1 shows the gunshot dataset used in our experiments. In the test dataset some files do not have gunshot inserted. Each file in the test dataset at most one gunshot is inserted. Each file in training and test set is 30 sec long and sampled at 44.1 KHz.

Table 5.1 Gunshot dataset

Training Dataset		**Test Dataset**
Type	**Number of Files**	**Number of Files**
Gunshot	202 (events)	89 (events)
City center	40	20
Park	37	10
Residential area	41	19
Beach	43	18
Metro station	41	22

Car crash dataset (Table 5.2) is taken from the research described in [35]. There are 56 files 3 min each in four folds. The 3 folds are used for training and last fold is used for testing. Total 204 car crashes are inserted in 56 background files at 15 dB SNR level. Audio files are sampled at 32 KHz. The detailed explanation of the dataset can be found in [35].

Table 5.2 Car crash dataset

Training Dataset		**Test Dataset**
Type	**Number of Files**	**Number of Files**
Car crash	150 (events)	54 (events)
Traffic	42	14

Scream dataset (Table 5.3) is taken from the research explained in [94]. There are 66 files for training and 20 files for testing. Each file is 3 min duration. The dataset contains sound files at 6 different SNR values, but we have taken only the files with SNR values of 15 dB. Table 5.3 shows the scream dataset properties.

Table 5.3 Scream dataset

Training Dataset		**Test Dataset**
Type	**Number of Files**	**Number of Files**
Scream	1881(events)	203 (events)
Various	66	20

Feature extraction is done as shown in Figure 5.6. First, we find the minimum length event signal in the dataset. This minimum length is the context length for that type of event. Each event signal in the training dataset is segmented according to this minimum length. Each segment is 40 ms windowed with 50% overlapping. From each 40 ms window, 40 MFCCs are calculated. Then concatenating these MFCCs we obtain one feature vector. Each event type has different context length. So each event sound has different size feature vector. These vectors have the output label “1”. The same procedure with the same segmenting and windowing sizes are applied to background sounds. The obtained vectors are labeled as “0”.

The minimum length of event signals (context lengths) and the number of MFCCs which are contained in their feature vector is shown in Table 5.4.

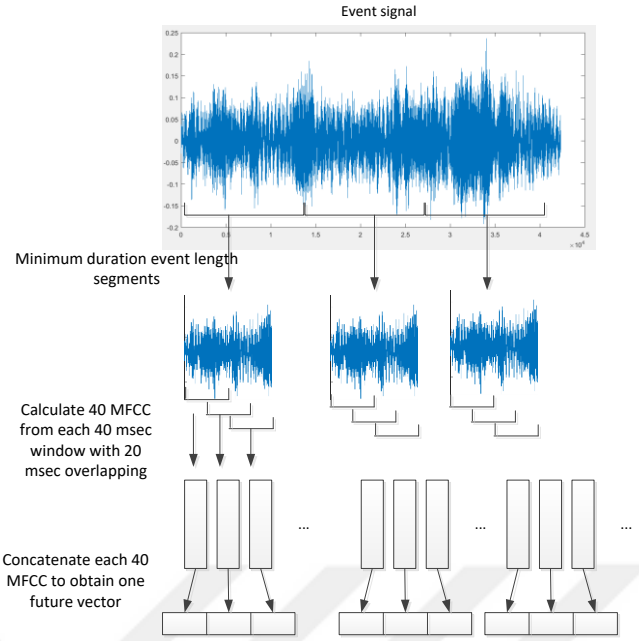


Figure 5.6 Feature Extraction

Table 5.4 Event sound feature properties

Event Name	Minimum Duration (ms)	Feature vector MFCC count
Gunshot	240	400
Car crash	711	1360
Scream	490	920

We used DNN as classification algorithm to recognize the sound events. Three DNN model has been developed for each event type.

Table 5.5 summarizes the hyper parameters of the DNNs used. We used rectified linear unit (RelU) as activation functions of hidden units and sigmoid function at the output unit. Figure 5.7 shows the DNN architecture used for car crash recognition.

Table 5.5 DNN hyper-parameters

Hyper –parameters	Gunshot	Car crash	Scream
# of layers	3	4	4
Learning rate	0.0075	0.085	0.075
Number of iterations	30000	10000	5000
Hidden Unit Activation	RelU	RelU	RelU
Output activation function	Sigmoid	Sigmoid	Sigmoid

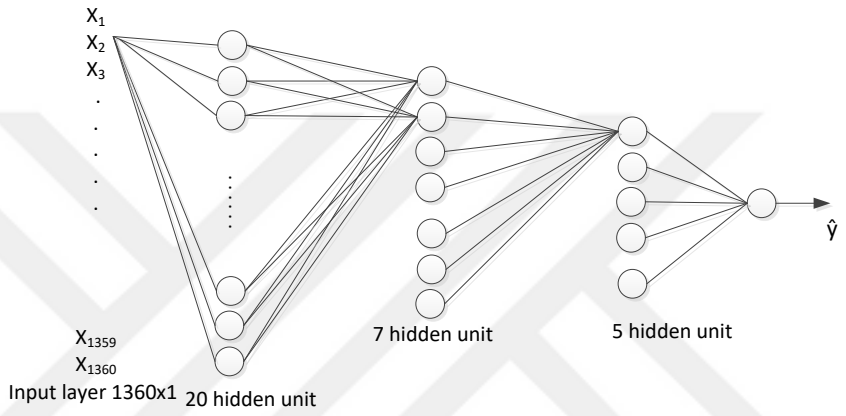


Figure 5.7 DNN model for car crash

We developed our own implementation of DNN for Matlab by porting the Python code developed during the course of DNN by Andrew NG on Coursera. The hyper-parameters are found manually making extensive tests. Grid search algorithms can be used to find best hyper-parameters, but our aim is to find hyper-parameters for an acceptable performance and make offline and online testing with these parameters.

5.6.2 Testing

Metrics:

For evaluation of the algorithm, event based F-score, error rate and other metrics such as accuracy, true positive and false positive rates are used. It can be misleading to look at one metric for performance criteria. The recognition rate can be very high but we should also look at error rate or false positive rate for correct evaluation of the

performance. The metrics alone are not meaningful for SED task. Most of the time there is a compromise between correct detections and false detections. When we try to increase TPs, FPs are increased too. The definitions for the metrics are in Table 5.6.

Table 5.6 Definitions of metrics used in SED [95]

Metric	Definition
True positives (TP)	Are detected events in the system output that have a temporal position overlapping with the temporal position of an event with the same label in the reference.
False positives (FP)	Are events in the system output that have no correspondence to an event with same label in the reference.
False negatives (FN)	Are events in the reference that have no correspondence to an event with same label in the system output.
True negatives (TN)	Are truly not detected events.
Insertions (I)	Are the number of events in system output that are not correct.
Deletions (D)	Are the number of reference events that were not correctly identified.
Substitutions (S)	Are events in the system output that have correct temporal position but incorrect class label.
Total events (N)	Are total number of events need to be detected.

Formulas for the metrics of SED are as follows [95]:

$$\text{Error rate (ER)} = \frac{S+I+D}{N}$$

$$\text{F-score} = \frac{2.P.R}{P+R}, \qquad P = \frac{TP}{TP+FP}, \quad R = \frac{TP}{TP+FN}$$

The above ER formulas defined in [95] is for polyphonic sound event detection, in this paper we deal with just monophonic sound event detection. For monophonic sound event detection where each sound clip contains one type of event sounds, we can write the error rate as follows:

$$\text{Error rate (ER)} = \frac{I+D}{N} = \frac{FP+FN}{N}$$

TPR which is also called sensitivity is the percentage of events correctly detected. FPR is percentage of non-event frames labeled as an event. Recognition rate is the percentage of sum of TPs and TNs to the sum of TPs, TNs, FPs and FNs. The formulas are as follows:

$$\text{TPR (sensitivity)} = \frac{TP}{TP+FN}, \text{FPR} = \frac{FP}{FP+TN}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Detailed explanation of these metrics can be found in [95].

Offline Tests:

In the audio surveillance area of SED, the developed algorithms will be used mainly for real-time applications. For example the method and the algorithms developed for road, home or elevator surveillance at the end will be used in real life application. So after testing algorithms on offline data it is necessary to switch more realistic scenarios for testing. In our case the same recorded audio data is played and it is captured by a microphone. Before deployment of the system, in this way a more realistic performance can be obtained and compared with the offline performance. Offline testing provides fast testing on large datasets while online data speed is the recording length of the dataset.



Figure 5.8 Offline SED

In Figure 5.8 schematic diagram of offline SED is seen. Audio file consists of the sounds to be detected and the background sounds. In our case the detected sounds are gunshots, car crashes and screams. The sounds to be detected are artificially embedded into background sounds. The background sounds are the sounds of environment in which the real-time detection will take place. The audio file is read

frame by frame and then features are extracted from these frames. Features are fed into SED algorithm for detection. SED algorithm labels the frame as normal audio event, gunshot, car crash or scream.

The test sound files are read in segments of minimum event length without overlapping. From these segments MFCC features are calculated as in training, and fed into DNN. The tests are done for each event type separately. For example we read a mixture file and feed it into a DNN model for finding the scream. Although some test sound files contain more than one type event such as car crash and scream we try to detect just scream.

A tolerance of 500 ms is used for detection of events such that we consider the detected event on time if the detected time is 500 ms before the event start time or 500 ms after event end time for offline detection. We used 1 sec tolerance for online detection. We compared the gunshot detection performance with the results of the DCASE 2017 task-2 results on the development dataset. Performance of gunshot detection is seen in Table 5.7 Performance comaparison of gunshot detection with other algorithms is shown in Table 5.8.

Table 5.7 Gunshot detection performance

Accuracy (%)	**TPR (%)**	**Error rate**	**F-score (%)**
99.9	89.7	0.52	77.4

Table 5.8 Comparison of gunshot detection with the baseline and some participants of DCASE 2017 Task-2 challange on the development dataset

	Error rate	**F-score(%)**
Proposed method	0.52	77.4
Baseline	0.69	57.4
Cakir et al. (evaluation dataset)	0.20	89.5
Dang et al.	0.37	79.2
Lim et al.	0.16	91.6

The performance of scream detection is seen in Table 5.9. Table 5.10 shows comparison results of scream detection with the results of the research in [94] and

[96]. Our proposed method recognition rate is 98.4% and outperforms [94] which is 87% and our accuracy is very close to the work done in [96].

Table 5.9 Performance of scream detection

Accuracy (%)	TPR (%)	Error rate	F-score (%)
98.4	87.6	0.57	75.4

Table 5.10 Accuracy comparison of secream detection with proposed method and other two studies on the same dataset

Proposed method (%)	Foggia et al. [94] (%)	Colangelo et al. (%)
98.4	87	98.5

Car crash detection results are seen in Table 5.11.

Table 5.11 Performance of car crash detection

Accuracy (%)	TPR (%)	Error rate	F-score (%)
98.4	77.7	0.35	81.5

The comparison of the car crash results with the work in [35] are seen in Table 5.12.

Table 5.12 Accuracy comparison of car crash detection with proposed method and the work in [35] on the same dataset

Proposed method (%)	Foggia et al. (%)
98.4	84.5

Online tests:

For online tests we prepared an automatic testing environment as in Figure 5.9, Figure 5.10. The microphone and speaker frequency response is important for the tests to be successful. The audio clips used in offline tests are sampled at 32 kHz for scream and car crash. The microphone and the speaker have to support the 16 kHz sound frequency for our algorithms as we used MFCC as feature and it depends on the audio clips frequencies. We used a microphone and speakers which supports the frequency range between 80 Hz and 20 kHz.

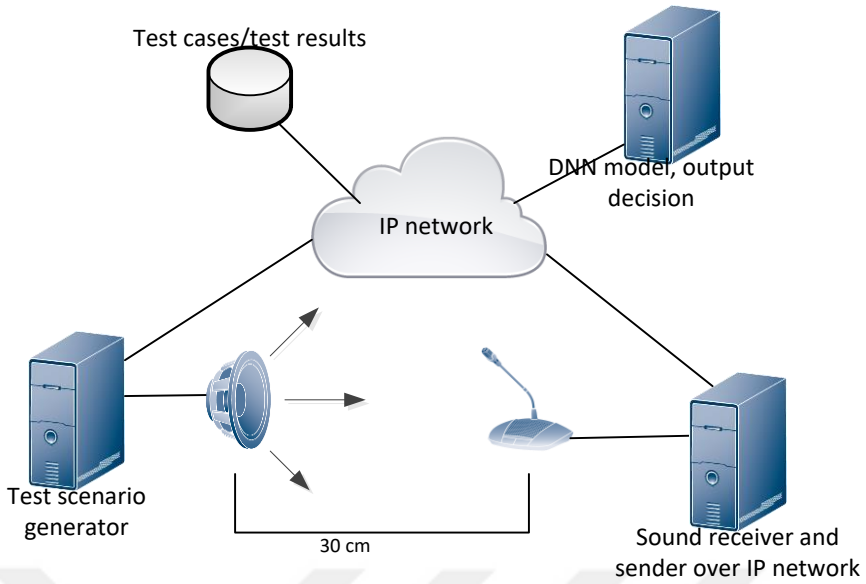


Figure 5.9 Online test for correct detection in room environment



Figure 5.10 Online testing environment at Atılım University

There are three pieces of software running on three different computers shown in Figure 5.9. The software which we call test scenario generator/player, reads the test sound files, writes the start of playing time of the sound file as offset and the start time of the events to the test cases table of the database. Then it plays each test sound file one by one. The other software captures the played sound with a microphone and sends the captured sound using user datagram protocol (UDP) over the local area

network (LAN) to the recognition software. The recognition software writes its decisions and the detection time to the test results table of the database. After inspecting test cases and test results the performance of online system is found.

This test environment installed three different size rooms and in anechoic room. In all three rooms we achieved almost the same results as offline case when the microphone to speaker distance is approximately 30 cm. When this distance is greater than 50 cm, the performance degraded significantly. In anechoic room the performance is again the same as offline case.

Online and offline test result comparisons are shown in the tables Table 5.13 and Table 5.14. Online results are the ones obtained in the room environment when the speaker to microphone distance is 30 cm.

Table 5.13 Comparison of offline and online scream detection

	TPR	**Error rate**	**F-score(%)**
Offline	87.6	0.57	75.4
Online	80.2	0.60	72.3

Table 5.14 Comparison of offline and online carcrash detection

	TPR	**Error rate**	**F-score(%)**
Offline	77.7	0.35	81.5
Online	75.4	0.40	75.7

Discussion:

During online tests we achieved the above results when the microphone is 30 cm apart from the speakers. If we increase the distance to 50 cm, the performance decreases to half. To prove that the distance problem is caused by reverberation, we repeated the online tests in an anechoic room. The anechoic room that we can use is a small one and we could put microphone and the speakers 2 m away at most. At this distance in anechoic room we measured the performance exactly the same as offline performance.

It is seen that the results are consistent as in distant speaker recognition (DSR) or another name, far-field speaker recognition. Performance results on DSR show that whenever speaker to microphone distant increases, recognition rates decrease and equal error rate (EER) increase [97]. In speaker recognition solutions such as reverberation compensation, feature warping or using multiple microphones can improve performance significantly [98].

In the field of hazardous SED to improve the performance we can use the followings:

- During training and model preparation we use event sounds degraded with reverberation.
- We record pure event sounds and then degrade it with different level of reverberation and then use these sounds for training.
- We can record the event sounds just at the same place where this SED application will be used. We then use these sounds for training.
- We can use event sounds without reverberation and during usage we can apply de-reverberation methods before recognizing the sound.

CHAPTER 6

EMERGING AUDIO SURVEILLANCE APPLICATIONS

6.1 Emerging Applications

According to Intelligent Data Corporation (IDC) which is global premier provider of the market intelligence, sound recognition is the key strategic technology that will be embedded in most connected devices offering artificial intelligence capabilities. The main applications (present and will emerge) can be divided into following categories [99]:

Home/Work place and Open Area/Campus security:

Audio surveillance can be used alone or by integrating to video surveillance systems. Audio surveillance for security can be used for home/work place security or for open area/campus security. For the first one there is no security infrastructure, so the audio data should be sent to cloud for processing and at the end for creating the alarms. For the latter one there is already a security infrastructure, so the audio data can be processed in the security room. In Figure 6.1 it is shown that audio data coming from microphone is processed in cloud or in the desktop computers of the security infrastructure.

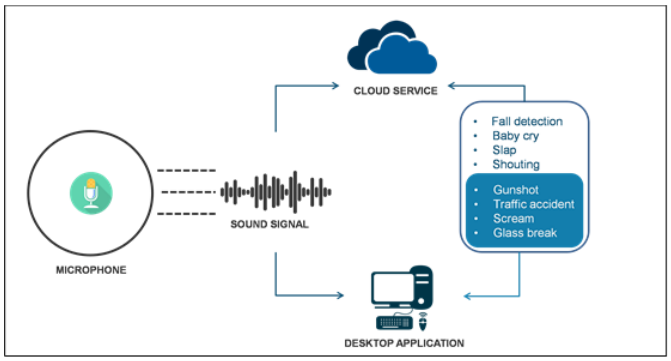


Figure 6.1 Audio surveillance for home/campus security

We will investigate audio surveillance for security for open area/campus in the next section in detail by integrating audio modality to current video surveillance systems.

Care:

A device detects changes to the home's audio profile through anomaly detection and takes actions. This device can also inform the nominated carer after detecting the sound of somebody crying or calling for help.

Pet care:

This is a major market for home owners who have pets at home during the day or evening. Home owners can be alerted if their pet is making a lot of noise, to enable them to review a live camera feed and then they can interact with their pet virtually.

Transportation:

The car's digital assistant detects threat sounds outside the car as well as distractions internally and alerts the driver to potentially dangerous or distracting situations. This can be the sound of an emergency vehicle or a car horn, or a vehicle reversing or emergency breaking.

Early traffic accident detection is important to save the lives of casualties. In European Union starting from 1 April 2018 all new cars will be sold with the E-Call system. E-Call is a system which calls the emergency services in case of a serious traffic accident.

In Figure 6.2 it is shown how E-Call is working. The E-Call system is made up of a series of in-vehicle sensors and processors fitted including in the airbags of the cars, linked with a communications device and a GPS tracker, all running thanks to a SIM card.

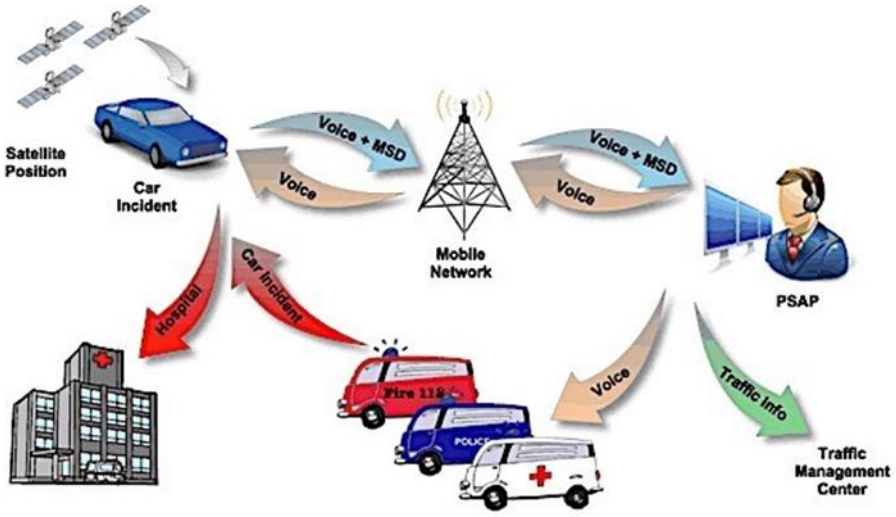


Figure 6.2 E-Call system

For the old cars a traffic accident detection system is also needed. We proposed the system shown in Figure 6.3 which uses the audio surveillance.

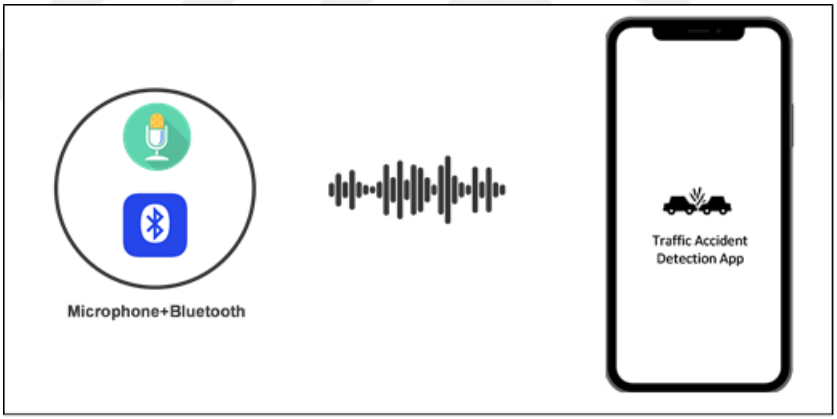


Figure 6.3 Proposed E-Call system for old cars

Entertainment:

While listening to music on the smart speaker, the device detects a noisy or distracting acoustic scene (from the kitchen, for example) and automatically adapts the music playback by changing various aspects of the sound spectrum to ensure the music cuts through. Similarly, the microphone pickup can be adjusted to take

account of the acoustic environment and make it easier to engage with a personal assistant. In home cameras designed to capture entertaining aspects of people's lives and family can automatically capture video or images when they detect the sound of laughter and automatically start to record.

Wellness and Wellbeing:

Sound recognition can be deployed in both smart homes and on personal devices such as wearables, hearables, and smartphones to improve the quality of life for occupants and users. For example, new parents can be alerted to the sound of their baby crying, with the home turning lights on to a low hue and playing a lullaby to help the baby back to sleep. Devices can also notify users if they are snoring and measure respiration during exercise.

Social:

At public places, sound recognition can improve the social environment and behavior by understanding the acoustic scene around them (such as a quiet restaurant or a busy coffee shop) and reacting accordingly (optimized microphone or speaker performance). Devices can also intelligently and seamlessly move the user from one experience to another without requiring manual intervention, such as hearables that can detect when the wearer is having a conversation while commuting or exercising, then can pause the music and increase transparency.

6.2 Integration of Audio Modality to Video Surveillance Networks for Open Area/Campus Security

6.2.1 Introduction

Modern video surveillance systems consist of a network of many video cameras. Constantly video camera systems are being installed for security reasons in prisons, elevators, automatic teller machines and more. Usually video cameras are connected to a display screen from which security personnel constantly monitor suspicious activity. As security personnel monitor multiple locations simultaneously, this

manual task is labor intensive and inefficient. These camera systems have some other drawbacks such that they have poor performance in adverse weather conditions or during night time. Therefore most of the time, some other sensors should accompany to video cameras. Although audio surveillance is in its early stage, there has been considerable amount of work in this area in the last decade. On the other hand currently there are no practical audio surveillance solutions for security on the market. In this section we tried to offer integration of audio surveillance into video surveillance networks. We followed all the processes of audio analysis system development by implementations.

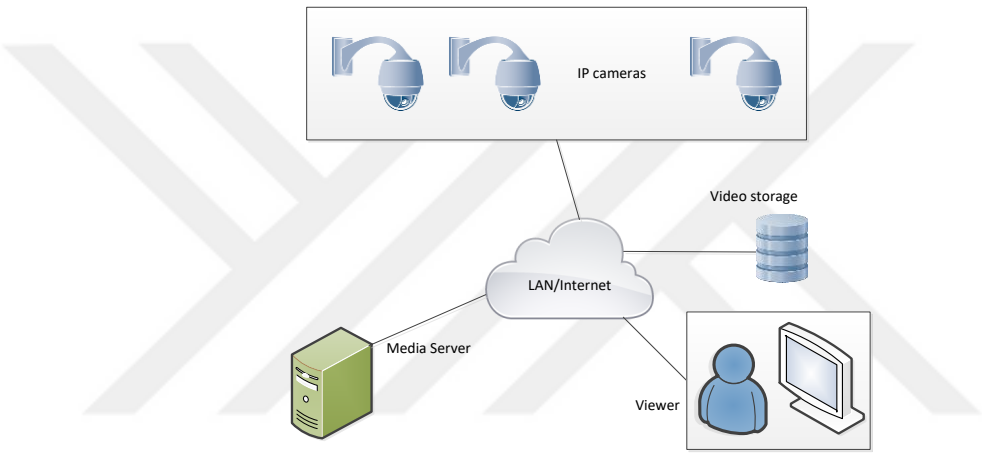


Figure 6.4 General video surveillance network

We explain the steps and give real examples to integrate audio surveillance into video surveillance systems. The difference of surveillance systems from the others is that they should respond in real-time. A general video based surveillance system is shown in Figure 6.4. This general system can be considered during the design of the development and test system of the audio surveillance system.



Figure 6.5 Control room of video surveillance network

In Figure 6.5 a typical control room of a video surveillance system for security is seen. It is clearly seen that there is an immediate need for this labor intensive and inefficient monitoring. Audio and video modality can be more efficient when used together. Alarms coming from audio surveillance can help the operator to watch the emergency screen. Audio surveillance can use the existing video surveillance infrastructure.

Figure 6.4 depicts general video surveillance camera network. The cameras here are Internet Protocol (IP) based. IP-based video surveillance systems have several operational and technological advantages over analog cameras, so the video surveillance network in this paper has IP-based cameras.

IP cameras in Figure 6.4 communicate with the media server in different ways. Transport Control Protocol (TCP) provides guaranteed delivery of packets by requiring acknowledgements by the receiver. Packets not acknowledged will be retransmitted. TCP can be beneficial especially for networks with inherent packet loss. While TCP induces more latency in the transport it is used for cameras that are only recorded and not typically viewed live. User Datagram Protocol (UDP) does not guarantee delivery and provides no facility for retransmission of lost packets. UDP transport is most suitable for networks with very little packet loss.

Media server is responsible from collecting the video from IP cameras and routing of the video to viewers. Media server distributes live and archived video streams to the viewers over the IP network simultaneously [100].

Viewers on the network request to see one or more camera video from the media server.

There is no hazardous SED solution alone or integrated with video surveillance system on the market to the best of our knowledge. The design steps required to integrate a hazardous SED to video surveillance systems are proposed as novelty in this study.

6.2.2 Latest Trends in Automatic Video Surveillance Systems

The paper [101] explains the historical development of remote surveillance systems. Surveillance systems have developed in three generations. The first generation of surveillance systems (1GSSs) used analogue equipment throughout the complete system [102]. Analogue closed-circuit television cameras (CCTV) captured the observed scene and transmitted the video signals over analogue communication lines to the central back-end systems, which presented and archived the video data [102]. The main challenge in the 1GSS is that it uses analogue techniques for image distribution and storage [103]. The second generation of surveillance systems (2GSSs) uses digital back-end components [102]. They enable real-time automated analysis of the incoming video data [102]. Automated event detection and alarms significantly increased the amount of simultaneously monitored data and the overall quality of the surveillance system [102]. The difficulty in the 2GSS is that it does not support robust detection and tracking algorithms, which are needed for behavioral analysis [103]. The third generation of surveillance systems (3GSSs) has finalized the digital transformation. In these systems, the video signal is converted into the digital domain at the cameras, which transmit the video data through a computer network, for instance a local area network. The back-end and transmission systems of a third-generation surveillance system have also improved their functionality [102].

The 3GSSs handle a large number of cameras, a geographical spread of resources, and many monitoring points. From an image processing view, they are based on the distribution of processing capacities over the network and the use of embedded

signal processing devices to achieve the benefits of scalability and potential robustness offered by distributed systems [102]. The current research of 3GSSs concentrates on distributed and centralized intelligence, data fusion, probabilistic reasoning frameworks, and multi-camera surveillance techniques [103].

During the evolution of 3GSS, different sensor modalities have been added to video surveillance network. EU project PRISMATICA [104], uses intelligent cameras, audio sensors and other methods to monitor different safety and security concerns in railways. The aim of the audio surveillance system in PRISMATICA is to enable the detection of distress calls such as “help”, “stop thief”, etc. or shouts during security problems such as assaults and thefts [104]. Another sensor, which is added to video surveillance networks, is thermal camera to increase visibility during nighttime [105].

3GSSs have also increased the video analytics. These intelligent activities are generalized as object detection, object recognition, tracking and behavior and activities analysis [105]. In [106], it is proposed a framework for smart surveillance system (S3) which is to enable different vendor video analytics into a large-scale system. This system has the functionalities such as; motion detection, directional motion detection, abandoned object alert, object removal and intentional camera movement or blinding.

In [107], a scalable 3GSS system is proposed for wide area monitoring. This system is outdoor, real-time, built with a cloud infrastructure. A four tier visualization subsystem, integrating text, image, streaming video and geographic context information is proposed. This system is an example of distributed intelligence. It provides the following functionalities:

- A robust parking space detection,
- Illegal parking events are handled by embedded illegal parking detection,
- The intrusion detection for climbing up,

- The embedded PTZ (Pan-Tilt-Zoom) controlling for moving object tracking component can monitor a moving target in a widely open area by automatic control of an embedded PTZ camera,

- Human face detection in dynamic scene component.

While 3GSSs continue to evolve, new surveillance technologies have emerged such as wireless and cloud based surveillance systems.

A wireless networking platform and video sensors to be used on this network are developed in [108]. In this work a video platform consisting a camera, microphone, a high speed application processor and Wi-Fi networking interfaces are designed. This platform is responsible for recording and transmitting video data. This video data is transmitted over a wireless network. The channel control of wireless network is done by ZigBee protocol and data is transmitted by Wi-Fi protocol to save energy. The main aim of this design is to overcome the infrastructure need of the 3GSSs.

Video Surveillance as a Service (VSaaS) is a new emerging technology. The paper [109] proposes a framework for a cloud-based multimedia surveillance system and highlights several research and technical issues. A cloud-based video surveillance system is proposed in [110] with emphasis on storage. Authors in [111] reported the design of a cloud-based scalable video recording system. Aside from video recording, their system also provided backup and monitoring features. They used Hadoop distributed file system for storing video recording. The authors consider cloud as a suitable platform to conduct video recording and analysis tasks. The authors in [112] reported their experience in designing video as a service. Their work concentrated on the deployment of software as a service platform for video surveillance and people re-identification in multi camera surveillance system.

The proposed cloud based surveillance system in [109] uses two types of content providers. One is the heterogeneous sensor devices, such as fixed cameras, IP cameras, and PTZ cameras, while the other is the crowd that reports security

incidents. All the tasks are done in the cloud servers and the users can get video and event information by connecting to Internet and the service.

6.2.3 Integration of Audio to Video Surveillance Network

The aim of our design is to integrate audio surveillance system into currently available video surveillance systems to decrease the burden of the work of the security personnel in the control room of video surveillance. In Section 6.2.2 we studied contemporary video surveillance systems. These systems mainly added video analytics to the 3GSSs to increase intelligence capability. In this work we try to add audio surveillance for security. Audio surveillance for security deals with detection of hazardous sound events such as gunshot, explosion, glass break, scream etc. By integrating hazardous SED with the video surveillance we expect that the video surveillance systems will be much more efficient. In the rest of this paper we will refer each hazardous SED application as application or audio application.

Although 3GSSs are widely deployed video surveillance solution, VSaaS and wireless systems are also widely studied solutions. But they are still lack of some basic requirements which are security, privacy, power management and cost. Because of this we based our deployment architecture to 3GSSs. 3GSSs are also divided as centralized or distributed intelligence. As a design choice, we design the audio integration for centralized intelligence. In the next subsections we describe the development stages of an audio surveillance system by giving real examples.

System Requirements:

At the beginning we shall make decisions for the system that we develop and determine system requirements. We shall decide to choose between centralized or distributed intelligence or use both architectures. At distributed intelligence, high speed computing devices receiving the microphone sound are deployed. These devices execute the SED application and send just the event if detected to central servers. This architecture is scalable if we will not deploy many applications, we can add nodes easily without affecting the performance of the system. This architecture is

also more suitable if the data sent over network is considered private. Drawbacks are that the intelligent devices are costly and we need many processing end nodes. Application scalability and flexibility are restricted. If we want to add a new application, it may be hard in terms of processing capability and work load to deploy a new application to the end nodes. On the other hand centralized intelligence provides more scalable and flexible applications. We do not need to control or configure end nodes for applications or we do not need to deploy applications to end nodes. Centralized architecture should have more powerful servers and need more communication bandwidth. Consequently, since our aim is to add scalable applications to video surveillance system, we can do this with the centralized intelligence architecture.

The developed system shall recognize the events such as gunshot, scream, car crash, tire skidding and this list shall be expanded. System shall give an alert if any of the events in this list is detected in real-time. The alarm shall inform the event name and the camera id on which the event can be seen. We will design our system for monophonic SED. SED where at most one simultaneous sound event is detected at a given time instance is called monophonic SED. On the other hand polyphonic SED is where multiple overlapping sound events can be detected at any given time instance [113].

The communication between microphones and the servers and all other communications shall be IP based. The existing 3GSS infrastructure will be utilized without additional needs.

According to security/privacy needs, the communication of audio data shall be encrypted. The audio data captured shall be stored or not stored. If audio data storage is required it shall be stored with the corresponding video data. Table 6.1 shows system requirements for audio modality addition to video surveillance network and Figure 6.6 shows general system architecture that will be developed.

Table 6.1 System requirements

Requirement	Design Choices	Choice
Architecture	Centralized Distributed Hybrid	Centralized
Communication	IP based TCP UDP RTP (Real Time Protocol)	UDP
Applications	Real-time, Offline (Gunshot, scream, car crash etc. detection)	Real-time
Security/privacy	Encrypted/not encrypted communication. Audio data will be stored or not stored.	Encrypted, Audio data will not be stored

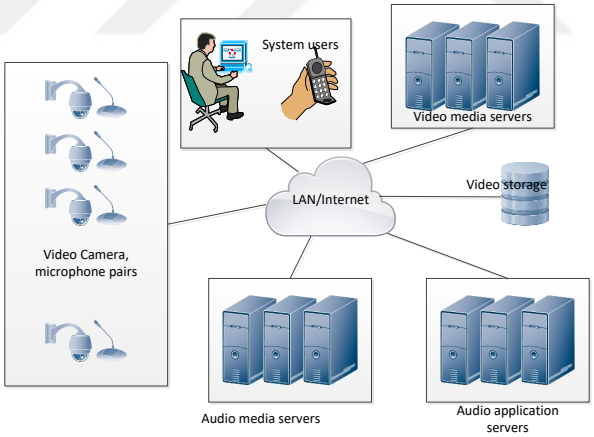


Figure 6.6 Audio integrated general system

System Development:

The main component of the system we will develop is the hazardous SED that will be added to the current video surveillance systems. So we will build our development strategy on the audio analysis system development methodology.

In Figure 6.7, the processes required for the development of an audio analysis system are seen [93]. The first process is the problem definition. By far we have defined the problem and we can skip to second process.

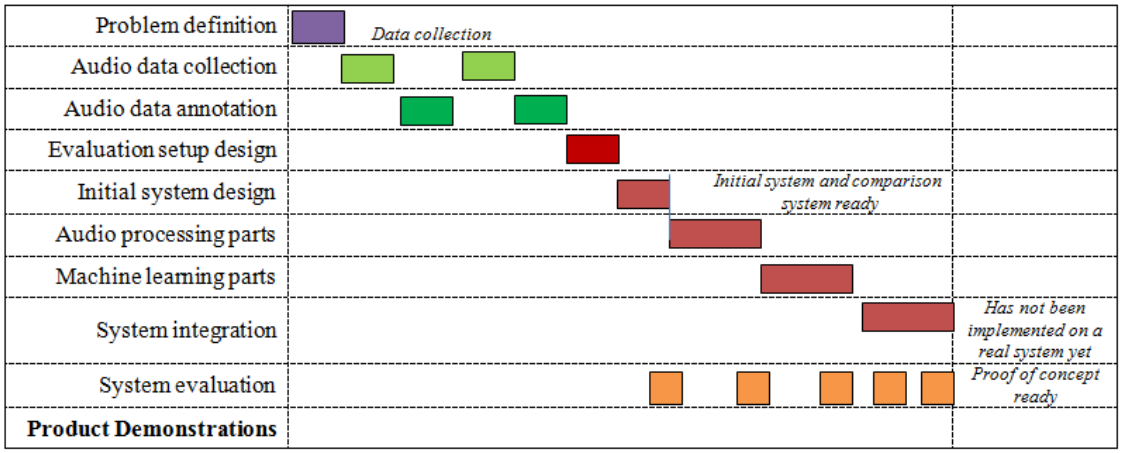


Figure 6.7 Process graph for audio analysis system development [93].

Audio data collection and audio data annotation:

Audio data is required for machine learning parts phase. To detect and recognize for example gunshots we should have gunshot sounds in advance to be able to prepare a machine learning model. Maybe one of the most difficult parts of the audio system design is to find quality sound clips for the events to be detected. We can handle these activities in three separate parts. These are data collection, annotation and database building.

In Section 5.6.1 we explained the resources to find annotated sound clips of hazardous events. We used gunshot, scream, car crash and the associated resources mentioned in Section 5.6.1 for the developed system. These sound clip datasets come with sound clips and an annotation file. In order to use these sound clips in the system design we should store them in a more convenient way. To store and retrieve the sound clips we designed a relational database explained in [1] and [2]. This database will be used during the lifecycle of the developed system. This database has the following properties:

- Sound clips and their properties are kept in a table such as sampling rate, quantization bit length and duration. These sound clips comprise the event sounds to be recognized.

- From the annotations, the isolated sounds of the events are extracted from sound clips and are kept in another table. In the machine learning parts phase, these extracted sound clips are used. Features extracted from these sound clips are labeled as “1”.

- Some background sound clips are stored in the database. These background sound clips can be city center, traffic, park, beach etc. The features are also extracted from these sound clips and labeled as “0”.

- Mixed sound clips including both event sounds and background sounds are stored in the database. These sound clips are used for testing the machine learning algorithms.

This database may be the most important part of the audio surveillance system, because it affects the performance of the system. During the lifecycle of the system this database will be used and always some sounds will be added.

Evaluation setup design:

Once the database is ready, evaluation setup is designed by selecting evaluation metrics, test dataset and by selecting appropriate comparison systems [93]. For evaluation of the algorithms event based F-score, error rate and other metrics such as accuracy, true positive and false positive rates are used. It can be misleading to look at one metric for performance criteria. The recognition rate can be very high but we should also look at error rate or false positive rate for correct evaluation of the performance. For the metrics see paragraph 5.6.2.

Initial system design:

At the system requirements subsection we made an architectural system design and defined the system components. Here we can detail the components and define how they work. Critical points are:

- A large number of microphone outputs are sent to server/s that is/are audio media server/s. These servers' network connection bandwidth and performance should be enough for these connections.

- The event detection applications will be scalable, so the architecture should support this. The scalability will be provided by the design seen in Figure 6.8. Audio media servers multicasts the each coming audio for each application using different UDP port number. In this way each application can be run on the same server or different physical server independently.

- If we or any vendor develops a new application, we can just start a new multicast on the audio media servers. We can add the new application on the existing servers or on a new physical server. Applications need to register multicast traffic of their application.

- Applications should be fast enough to respond in real-time.

The initial system design to support these needs is seen in Figure 6.8.

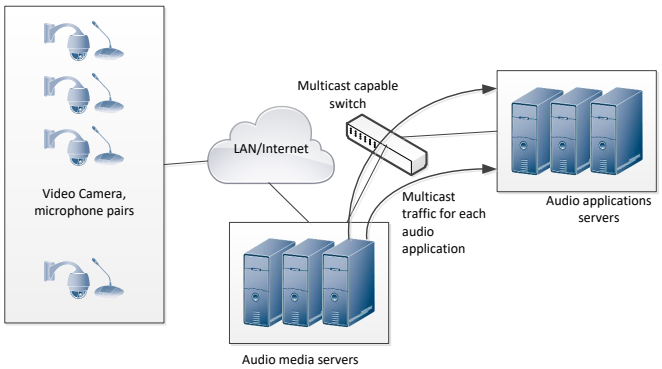


Figure 6.8 Initial system design

Audio processing parts:

Audio is prepared and processed for machine learning algorithms in the audio processing phase of the overall system design [93]. We can divide this stage as audio pre-processing and feature extraction. The pre-processing is adjusting audio channel information, resampling according to application needs, noise reduction or sound source separation.

Audio pre-processing is done during development, evaluation and during usage. While the audio data is collected from various sources, it is most likely captured in non-uniform recording settings, with variations in the amount of captured audio channel, and used sampling frequency [93]. We have to convert these to the actual settings which will be used at the deployment. In deployment if we use microphones supporting up to 16 kHz we should resample our audio data in the database to 32 kHz according to the Nyquist theorem.

Feature extraction is the second part of the audio processing. Frequency response of the microphones is important, because we use the Mel-Frequency Cepstral Coefficients (MFCC) as the feature. Feature extraction is done as explained in section 5.6.1

Machine learning parts:

Machine learning can be divided into three stages. They are training or learning, testing and usage stages. We explained the detail of machine leraning in the section 5.1.

System evaluation:

The evaluation of the system is based on the test data and reference annotations assigned to it, and using a metric related to the target application. Ideally the system should be evaluated in scenarios which match the target application as much as possible, to get a realistic performance estimation. During the development, the evaluation is commonly done in steps by gradually increasing how closely the

scenario matches the target application to better isolate factors affecting the performance [93]. We can evaluate our system in three parts. First part is the heart of the system and system success depends on it. It is the DNN model. DNN model performance in audio surveillance is explained in [114]. Second part is the audio transport on the network and the third part is the parallel processing of the developed DNN (SED) models which we named before applications or audio applications.

For the DNN model evaluations we prepared offline and online test setups. For offline and online evaluation of the algorithms the steps and setup seen in Section 5.6.2 are used.

The prepared online evaluation setup seen in Figure 5.9 is partially give opportunity to test the audio transport over the network. By increasing the number of microphones and connected computers and distributing different places, we can make more realistic evaluation which represents the final product.

Discussion:

Modern video surveillance systems consist of many cameras. They work on computer networks and it is easy to add cameras to existing networks. Increasing speed of computer networks permits many cameras on the same network. Monitoring of this large number of cameras is not an easy task. Although there are now practical implementations of video analytics, it does not solve the need of intelligence in the area. We need more intelligence to get the most out of video surveillance systems. Video surveillance systems need solutions for real-time response to incidents. One of the most studied areas on this subject is the hazardous SED. Lately DNNs have improved performance on SED. However, to the best of our knowledge there is no SED solution alone or integrated with video surveillance networks on the market. The simulations of the designed hazardous SED employing a DNN structure offers very high detection performances more than 98.4% for gunshot, scream and car crash. Our design is a scalable architecture which permits integration of applications from different vendors.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Summary of Completed Work

Firstly we developed a database which has a convenient structure and tools for impulsive sound detection. Then we found the data for gunshot, scream and car crash detection. The sounds we loaded to database were from other research. We gave detailed explanation of the database and the data addresses for other researchers.

We made an extensive research on impulsive sound detection algorithms. Firstly, in literature we considered impulsive noise detection algorithms for impulsive sound detection. We made taxonomy of impulsive sound/noise detection algorithms to find the algorithms for online real-time detection. We showed that mostly used energy based detectors can be outperformed by impulsive noise detection algorithms. We put forward three models for the detection and recognition of impulsive sounds.

At the recognition subsystem we used state of the art machine learning algorithm which is DNN. We prepared automatic test environment for offline and online tests. Test results show that there is a big difference between offline and online test if the speaker and microphone distance is greater than 30 cm. We realized that the reason of this is because of reverberation. This is a known problem in DSR. We proposed methods to overcome this problem, otherwise the algorithms developed for SED for surveillance applications are useless in real environments.

In the last chapter we developed step by step application to integrate the audio modality to video surveillance networks. If the DNN models we developed for hazardous SED is integrated into video surveillance networks, the efficiency of this systems are improved significantly. This is just an application of hazardous SED.

7.2 Future Work

The developed database is capable of expansion. We can load other impulsive sounds such as explosions, glass break, siren, etc. This database can be accessed through Internet and used by other researchers.

We provided three methods for impulsive sound recognition and actually used the third one in our tests. The other methods can be studied if a faster system is needed.

We proposed reverberation cancelling methods for better real environment applications, but we couldn't implement these methods. These methods can be studied and a comparison between them can be made.

Although we design a high level architecture for integration of audio modality to video surveillance system we did not implement it completely. A demo version of this system can be implemented.

REFERENCES

[1] Arslan, Y. & Canbolat, H., “A sound database development for environmental sound recognition”, Signal Processing and Communications Applications Conference (SIU), 25th. 2017.

[2] Arslan, Y., Tanıs, A. & Canbolat H., “A Relational Database Model and Tools for Environmental Sound Recognition”, ASTES Journal, *Volume 2*, Issue 6, Page No 145-150. 2107.

[3] Dufaux, A., “Detection and recognition of Impulsive Sound Signals”, Ph.D. Thesis, University of Neuchatel, 2001.

[4] Lojka, M., Pleva, M., Kiktova, E., Juhar, J. and Cizmar, A., “Ear-tuke: The acoustic event detection system”, Multimedia Communications, Services and Security, vol. 429 of Communications in Computer and Information Science, pp. 137-148. 2014.

[5] http://www.indect-project.eu/

[6] Lopatka K. & Czyzewski, A., “Acceleration of decision making in sound event recognition employing supercomputing cluster”, Information Sciences, 285:223–236. 2013.

[7] Pillos, A., Alghamidi, K., Alzamel, N., Pavlov, V. and Machanavajhala, S., “A Real Time Environmental Sound Recognition System”, Detection and Classification of Acoustic Scenes and Events 2016, Budapest, Hungary. 2016.

[8] Ellis, D.P.W., “Prediction-driven computational auditory scene analysis”, Ph.D. Thesis, Massachusetts Institute of Technology, 1996.

[9] Brown, G.J. & Cooke, M.P., “Computational auditory scene analysis, Computer Speech and Language", 8, 297-336. 1994.

[10] Mellinger, D. K., “Event Formation and Separation in Musical Sounds”, Ph.D. Thesis, Department of Music, Stanford University, CA. 1991.

[11] Bregman, A., “Auditory scene analysis”, USA: MIT Press, 1990.

[12] Eronen, A., Peltonen, V., Tuomi, J., Klapuri, A., Fagerlund, S., Sorsa, T., Lorho, G., and Huopaniemi, J., “Audio-based context recognition”, IEEE Trans. Audio, Speech, Lang. Process., vol. 14, no. 1, pp. 321–329, Jan. 2006.

[13] Temko, A., “Acoustic Event Detection and Classification”, Ph.D. dissertation, Universitat Politecnica De Catalanya. 2007.

[14] Peltonen, V., “Computational auditory scene recognition”, Ph.D. dissertation, Tampere University of Technology, Finland. 2001.

[15] Benzeghiba, M., De Mori, R., Deroo, O., Dupont, S., Erbes, T., Jouvet, D., Fissore, L., Laface, P., Mertins, A., Ris C. and et al., “Automatic speech recognition and speech variability: A review”, Speech Communication, vol. 49, no. 10-11, pp. 763–786. 2007.

[16] Reynolds, D.A., “An overview of automatic speaker recognition technology”, IEEE International Conference on Acoustics, Speech, and Signal Processing, Proceedings, (ICASSP'02) vol. 4. 2002.

[17] Reynolds D.A. & Torres-Carrasquillo, P., “Approaches and applications of audio diarization”, Proceedings of ICASSP, pp. 953–956. 2005.

[18] Muthusamy, Y.K., Barnard, E., & Cole, R.A., “Automatic language identification: A review/tutorial”, IEEE Signal Processing Magazine, vol. 11, no. 4, pp. 33–41. 1994.

[19] Ahmed, T., Uppal, M., and Muhammad, A., “Improving Efficiency and Realibility of Gunshot Detection Systems”, IEEE, ICASSP. 2013.

[20] Gerosa, L., Valenzise, G., Tagliasacchi, M., Antonacci, F. and Sarti, A., “Scream and Gunshot Detection in Noisy Environments”, 15th European Signal Processing Conference (EUSIPCO-07), Sep.3-7, Poznan, Poland. 2007.

[21] Freire, I.L. & Apolinario Jr, J.A., “Gunshot detection in noisy environment”, Proceeding of the 7th International Telecommunications Symposium, Manaus, Brazil. 2010.

[22] Biondi, R., Dys, G., Ferone, G., Renard, T. and Zysman, M., "Low Cost Real Time Robust Identification of Impulsive Signals", International Journal of Computer Electrical Automation Control and Information Engineering, vol. 8, no. 9, pp. 1653-1656. 2014.

[23] Thumwarin, P., Matsuura, T. and Yakoompai, K., "Audio forensics from gunshot for firearm identification", Proc. IEEE 4th Joint International Conference on Information and Communication Technology Electronic and Electrical Engineering Tailandia, pp. 1-4. 2014.

[24] Chu, S., Narayanan, S., Kuo, C.-C.J. and Mataric, M.J., "Where amI? Scene recognition for mobile robots using audio features", IEEE Int.Conf. on Multimedia and Expo. IEEE, 885–888. 2006.

[25] Yamakawa, N., Takahashi, T., Kitahara, T., Ogata, T. and Okuno, H.G., "Environmental sound recognition for robot audition using Matching-Pursuit", in K.G. Mehrotra, C.K. Mohan, J.C. Oh, P.K. Varshney & M. Ali (Eds), Modern Approaches in Applied Intelligence, Berlin Heidelberg: Springer, 1–10. 2011.

[26] Chen, J., Kam, A.H., Zhang, J., Liu, N. and Shue, L., "Bathroom activity monitoring based on sound", in H.W. Gellersen, R.Want, & A. Schmidt (Eds), Pervasive Computing, Berlin Heidelberg: Springer, 47–61. 2005.

[27] Vacher, M., Portet, F., Fleury, A. and Noury, N., "Challenges in the processing of audio channels for ambient assisted living", 12th IEEE Int. Conf. on e-Health Networking Applications and Services (Healthcom), 330–337. 2010.

[28] Wang, J.-C., Lee, H.-P., Wang, J.-F. and Lin, C.-B., "Robust environmental sound recognition for home automation", IEEE Transactions on Automation Science and Engineering, 5 (1), 25–31. 2008.

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

[30] Bardeli, R., Wolff,D., Kurth, F., Koch,M., Tauchert, K.-H., Frommolt, K.-H., "Detecting bird sounds in a complex acoustic environment and application to bioacoustic monitoring", Pattern Recognit. Lett., 31 (12), 1524–1534. 2010.

[31] Weninger, F. & Schuller, B., "Audio recognition in the wild: static and dynamic classification on a real-world database of animal vocalizations", IEEEInt.Conf. on IEEE Acoustics, Speech and Signal Processing (ICASSP), 337–340. 2011.

[32] Raty, T., "Survey on contemporary remote surveillance systems for public safety", Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 40(5):493–515. 2010.

[33] Fong A.C.M. & Hui, S. C., "Web-based intelligent surveillance system for detection of criminal activities", Comput. Control Eng. J., vol. 12, no. 6, pp. 263–270. 2001.

[34] Collins, R. T., Lipton, A. J., Fujiyoshi, H. and Kanade, T., "Algorithms for cooperative multisensor surveillance", Proc. IEEE, vol. 89, no. 10, pp. 1456–1477, Oct. 2001. doi: 10.1109/5.959341.

[35] Foggia, P., Petkov, N., Saggese, A., Strisciuglio, N. and Vento, M., "Audio surveillance of roads: a system for detecting anomalous sounds", IEEE Transactions on Intelligent Transportation Systems. 2016.

[36] Rouas, J.-L., Louradour, J. and Ambellouis, S., "Audio events detection in public transport vehicle", Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC '06), pp. 733–738, Toronto, Canada, September. 2006.

[37] Radhakrishnan, R. & Divakaran, A., "Systematic acquisition of audio classes for elevator surveillance", Image and Video Communications and Processing, vol. 5685 of Proceedings of SPIE, pp. 64–71, March. 2005.

[38] Harma, A., Mckinney, M. F. and Skowronek, J., "Automatic surveillance of the acoustic activity in our living environment", Proceedings of the IEEE International

Conference on Multimedia and Expo (ICME ’05), pp. 634–637, Amsterdam, July. 2005.

[39] Atrey, P. K., Maddage, N. C. and Kankanhalli, M. S., “Audio based event detection for multimedia surveillance”, Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’06), vol. 5, pp. 813–816, Toulouse, France, May. 2006.

[40] Lupatka, K., Zwan, P. and Czyzewski, A., “Dangerous sound event recognition using support vector machine classifiers”, Advances in Multimedia and Network Information System Technologies Volume 80 of the series Advances in Intelligent and Soft Computing pp 49-57. 2010.

[41] Okamoto, A., Hayashida, K., Nakayama, M. and Nishiura, T., “A detection of danger sounds based on variable-state hidden Markov models”, Signal Processing in Acoustics. 2013

[42] Demir, F., Lu, H., Sengur, A. and Amiriparian, S., “Compact Bilinear Deep Features For Environmental Sound Recognition”, IDAP 2018, International Conference on Artificial Intelligence and Data Processing, Malatya, Turkey. 2018.

[43] Chu, S., Narayanan, S. and Kuo, C.-C. J., “Environmental sound recognition with time– frequency audio features”, IEEE Trans. Audio, Speech, Lang. Process., vol. 17, no. 6, pp. 1142–1158. 2009.

[44] Lojka, M., Pleva, M., Kiktova, E., Juhar, J. and Cizmar, A., “Efficient acoustic detector of gunshots and glass breaking”, Multimedia Tools and Applications, 75(17):10441–10469. 2016.

[45] Gygi, B. & Shafiro, V., “Development of the Database for Environmental Sound Research and Aplication (DESRA): Design, Functionality and Retrieval Considerations”, EURASIP Journal on Audio, Speech, and Music Processing. 2010.

[46] Van Grootel, M.W.W., Andringa, T.C. and Krijnders, J.D., "DARES-G1: Database of Annotated Real-world Everyday Sounds", Proceedings of the NAG/DAGA Meeting Rotterdam. 2009.

[47] Dickerson, K., Gaston, J. R. and McCarty-Gibson, S., "Parameterizing Sound: Design Considerations for an Environmental Sound Database", Army Research Laboratory. 2015.

[48] Nakamura, S., Hiyane, K., Asano, F., Nishiura, T. and Yamada, T., "Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition", LREC. 2000.

[49] DCASE 2016 WEB site available at: http://www.cs.tut.fi/sgn/arg/dcase2016/ (27.09.2017 last accessed)

[50] DCASE 2017 WEB site available at: http://www.cs.tut.fi/sgn/arg/dcase2017/ (27.09.2017 last accessed)

[51] Piczak, K. J., "ESC: Dataset for environmental sound classification", Proceedings of the ACM International Conference on Multimedia. 2015.

[52] Salamon, J., Jacoby C.and Pablo Bello, J., "A Dataset and Taxonomy for Urban Sound Research", Proceedings of the 22nd ACM international conference on Multimedia, November 03-07. 2014.

[53] Arslan Y., Tanıs A. and Canbolat H., "A Relational Database Model and Tools for Environmental Sound Recognition", ASTES Journal, Volume 2, Issue 6, Page No 145-150. 2017.

[54] Gaver, W. W., "What in the world do we hear? An ecological approach to auditory event perception", Ecological Psychology, vol. 5, no. 1, pp. 1–29. 1993.

[55] Brown, A. L., Kang, J. and Gjestland, T., "Towards standardization in soundscape preference assessment", Applied Acoustics, 72(6):387-392. 2011.

[56] Dennis, J. W., “Sound Event Recognition in Unstructured Environment Using Spectrogram Image processing”, Ph.D. Thesis, Nanyang Technological University. 2014.

[57] Park, S., Lee, Y., Han, D.K. and Ko, H., “Subspace projection cepstral coefficients for robust acoustic event detection”, Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). 2017.

[58] Rabaoui, A., Davy, M., Rossignol, S., Lachiri, Z. and Ellouze, N., “Using one-class SVMs and wavelets for audio surveillance systems”, IEEE Trans. Inf. Forensic Security. 2007.

[59] Kauppinen, I., “Methods for Detecting Impulsive Noise in Speech and Audio Signals”, Digital Signal Processing. 2002.

[60] Godsill S.J. & Rayner, P.J.W., “Digital Audio Restoration - a statistical model based approach”, 1998.

[61] Crocco, M., Cristani, M., Trucco, A. and Murino, V., “Audio surveillance: A systematic review”, CoRR, vol. abs/1409.7787. 2014.

[62] Machmer, T., Swerdlow, A., Kroschel, K., Moragues, J. and at all, “Robust Impulsive Sound Source Localization By Means of an Energy Detector for Temporal Alignment and Pre-Classification”, European Signal Processing Conference, 17th. 2009.

[63] Vacher, M., Istrate, D., Besacier, l., Serignat, J. and Castelli, E., “Life Sounds Extraction and Classification in Noisy Environment”, Proc. IASTED International Conference on Signal & Image Processing. 2003.

[64] Jorge, M., Arturo, S., Vergara, L. and Gosálbez, J., “Improving detection of acoustic signals by means of a time and frequency multiple energy detector”, IEEE Signal Processing Letters, vol. 18, n. 8, pp. 458-461. 2011.

[65] Azlan, M., Cartwright, I., Jones, N., Quirk, T. and West, G., "Multimodal monitoring of the aged in their own homes", Proceedings of the 3rd International Conference on Smart Homes and Health Telematics (ICOST'05). 2005.

[66] Smeaton, A. F. and McHugh, M., "Towards event detection in an audio-based sensor network", in Proc. 3rd Int. Workshop Video Surveill. Sens. Netw. (VSSN), pp. 87-94, Nov. 2005.

[67] Morton Jr., K.D., "Bayesian Techniques for Adaptive Acoustic Surveillance", Ph.D. Thesis, Duke University. 2010.

[68] Morton Jr., K.D., "Bayesian Detection of Acoustic Muzzle Blasts", Proc. of SPIE Vol. 7305 730511-1. 2009.

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

[70] Cristani, M., Bicego, M. and Murino, V., "On-line adaptive background modelling for audio surveillance", Proceedings of the 17th International Conference on Pattern Recognition, vol. 2, pp. 399 – 402. 2004.

[71] Ito, A., Aiba, A., Ito, M. and Makino, S., "Detection of abnormal sound using multi-stage GMM for surveillance microphone", Information Assurance and Security,. IAS '09. Fifth International Conference on, vol. 1, pp. 733 –736. 2009.

[72] Vaseghi, S. & Rayner, P., "Detection and suppression of impulsive noise in speech communications systems", Proc. Inst. Elec. Eng. Commun. Speech Vision, pp. 38-46, Feb. 1990.

[73] Esquef, P.A.A., Karjalainen, M. and Vlimki, V., "Detection of clicks in audio signals using warped linear prediction", 14th IEEE Int. Conf. Digital Signal Process. (DSP-02). 2002.

[74] Roch, M.A., Klinck, H., Baumann-Pickering, S., Mellinger, D.K. and Qui, S., "Classification of echolocation clicks from odontocetes in the Southern California Bight", Acoustical Society of America. 2011.

[75] Kandia, V. & Stylianou, Y., "Detection of Sperm Whale Clicks Based On the Teager – Kaiser Energy Operator", Applied Acoustics 67 pp. 1144–1163. 2006.

[76] Oudre, L., "Automatic detection and removal of impulsive noise in audio signals", Image Processing On Line. Preprint. February. 2014.

[77] Murphy, J., & Godsill, S., "Joint Bayesian removal of impulse and background noise", Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 261-264. http://dx.doi.org/10.1109/ICASSP.2011.5946390. 2011.

[78] Stallmann, C.F. & Engelbrecht, A.P., "Predictive Noise Detection in Gramophone Records", International Journal of Signal Processing Systems Vol. 4, No. 6, December. 2016.

[79] Niedzwiecki, M. & Ciołek, M., "Elimination of impulsive disturbances from archive audio signals using bidirectional processing", IEEE Trans. Audio, Speech, Lang. Process., vol. 21, no. 5, pp. 1046–1059, May. 2013.

[80] Niedzwiecki, M. & Ciolek, M., "New Semicausal and Noncausal Techniques for Detection of Impulsive Disturbances in Multivariate Signals with Audio Applications", IEEE Transactions on Signal Processing, Vol. 65, No:15, August. 2017.

[81] Niedzwiecki, M., Ciołek, M. and Cisowski, K., "Elimination of impulsive disturbances from stereo audio recordings using vector autoregressive modeling and variable-order Kalman filtering", IEEE Trans. Audio, Speech, Lang. Process., vol. 23, no. 6, pp. 970–981, Jun. 2015.

[82] Ciolek, M. & Niedzwiecki, M., “Detection of impulsive disturbances in archive audio signals”, Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). 2017.

[83] Ruhland, M., Bitzer, J., Brandt, M. and Goetze, S., “Reduction of Gaussian, supergaussian, and impulsive noise by interpolation of the binary mask residual”, IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 10, pp. 1680 – 1691. 2015.

[84] Vaidyanathan, P. P. “The Theory of Linear Prediction”, 2008.

[85] http://legacy.spa.aalto.fi/software/warp/ (last accessed: 11/08/2017)

[86] Wolfe, P.J., Godsill, S.J. and Ng, W.J., “Bayesian variable selection and regularization for time-frequency surface estimation”, Journal of the Royal Statistical Society, Series B, vol. 66, no. 3, pp. 575–589. 2004.

[87] Wolfe, P.J. & Godsill, S.J., “Interpolation of missing data values for audio signal restoration using a Gabor regression model”, Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 517-520, Mar. 2005.

[88] http://ltfat.sourceforge.net/ (last accessed: 11/08/2017).

[89] Al-Manie, M.A. & Wang, W.J., “Inverse Gabor Transform for Speech Enhancement”, Acoustics Australia, Vol. 41, No. 3, December. 2013.

[90] Mesaros, A., Heittola, T., Diment, A., Elizalde, B., Shah, A., Vincent, E., Raj, B. and Virtanen, T., “DCASE 2017 challenge setup: tasks, datasets and baseline system”, Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017), November. 2017.

[91] Martin, A., Doddington, G., Kamm, T., Ordowski, M. and Przybocki, M., “The DET Curve in Assessment of Detection Task Performance”, DTIC Document, Tech. Rep. 1997.

[92] https://martechtoday.com/how-machine-learning-works-150366. (02.11.2018 last accessed).

[93] Virtanen, T., Plumbley, M. and Ellis, D., "Computational Analysis of Sound Scenes and Events", Springer, 21 Sep. 2017.

[94] Foggia, P. Petkov, N., Saggese, A., Strisciuglio, N. and Vento, M., "Reliable detection of audio events in highly noisy environments", Pattern Recognition Letters, vol. 65, pp. 22-28. 2015

[95] Mesaros, A., Heittola, T. and Virtanen, T., "Metrics for Polyphonic Sound Event Detection", Applied Sciences, vol. 6, no. 6, p. 162. 2016.

[96] Colangelo, F., Battisti, F., Carli, M. and Neri, A., "Enhancing audio surveillance with hierarchical recurrent neural networks", IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2017, 14th, Sept. 2017.

[97] Nematollahi, M.A, & Al-Haddad, S.A.R., "Distant speaker recognition: an overview, International Journal of Humanoid Robotics", pp. 1550032. 2015.

[98] Jin, Q., Schultz, T. and Waibel, A., "Far-Field Speaker Recognition", IEEE TASLP, vol. 15, no. 7, pp. 2023-2032, Sept. 2007.

[99] Jeronimo, F. https://content.audioanalytic.com/hubfs/IDC-exec-brief-sound-recognition-web.pdf?t=1513775685614&utm_campaign=IDC%20report&utm_source=hs_automation&utm_medium=email&utm_content=59499157&_hsenc=p2ANqtz-91Nndjj-88m-gVlUdqMlskQY9a4wNJI_uGpwgo3HPPCP_7p4y4r3c0prwMMG6WVtRBlRdxtE_EEMaZIBOQBJiX4C0efg&_hsmi=59499157, December. 2017.

[100] Cisco IP Video Surveillance Design Guide, August. 2009.

[101] Raty, T.D., "Survey on contemporary remote surveillance systems for public safety", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol.99, pp.1-23. 2010.

[102] Bramberger, M., Doblander, A., Maier, A., Rinner, B. and Schwabach, H., "Distributed embedded smart cameras for surveillance applications", Computer, vol. 39, no. 2, pp. 68–75, doi: 10.1109/MC.2006.55. 2006.

[103] Valera, M. & Velastin, S. A., "Intelligent distributed surveillance systems: A review", IEE Proc.-Vis. Image Signal Process., vol. 152, no. 2, pp. 192–204, Apr., doi: 10.1049/ip-vis: 20041147. 2005.

[104] Lo, B.P.L., Sun, J. and Velastin, S.A., "Fusing Visual and Audio Information in a Distributed Intelligent Surveillance System for Public Transport Systems", Acta Automatica Sinica, 29, 393-407. 2003.

[105] Gade, R. & Moeslund, T.B., "Thermal cameras and applications: A survey", Machine Vision & Applications 25, 245-262. 2014.

[106] Tian, Y.L., Brown, L., Hampapur, A., Lu, M., Senior, A. and Shu, C.F., "IBM smart surveillance system (S3): event based video surveillance system with an open and extensible framework", Machine Vision and Applications, vol.19 no.5-6, pp.315-327. 2008.

[107] Wang Y.-K., Fan C.T. and Huang, C.R., "A large scale video surveillance system with heterogeneous information fusion and visualization for wide area monitoring", Proceedings of the 8th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP ′12). 2012.

[108] Cho, H., Baek, Y. and Kyung, C.M., "Wireless Video Sensor Network Platform and Its Application for Public Safety", In: Proc. IEEE Int. Conf. HPCC, CSS, ICESS, pp. 743-746. Aug. 2014.

[109] Hossain, A.A., "Framework for a cloud-based multimedia surveillance system", Int. J. Distrib. Sensor Netw., vol. 2014.

[110] Rodrıguez-Silva, D.A., Adkinson-Orellana L., Gonzalez-Castano F.J., Armino-Franco, I., Gonzalez-Martinez, D., "Video surveillance based on cloud storage",

Proceedings of the IEEE 5th International Conference on in Cloud Computing (CLOUD'12), pp. 991–992. 2012.

[111] Lin, C.F., Yuan, S.M., Leu, M.C. and Tsai, C.T., "A framework for scalable cloud video recorder system in surveillance environment", Proceedings of the 9th International Conference on Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC '12), pp. 655–660. 2012.

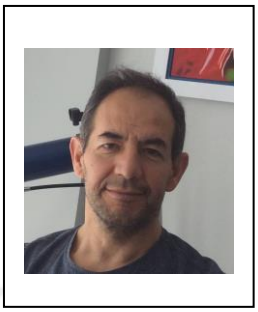
[112] Cucchiara, R., Prati, A. and Vezzani, R., "Designing video surveillance systems as services", Proceedings of the 2nd Workshop on Video Surveillance Projects in Italy (VISIT '11). 2011.

[113] Cakir, E., Parascandolo, G., Heittola, T., Huttunen, H., Virtanen, T., "Convolutional recurrent neural networks for polyphonic sound event detection", IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 25, no. 6, pp. 1291–1303. 2017.

[114] Arslan, Y. & Canbolat, H., "Performance of Deep Neural Networks in Audio Surveillance", 6th International Conference on Control Engineering and Information Technologies, 25-27 October 2018, Istanbul, Turkey.2018.

CURRICULUM VITAE

PERSONAL INFORMATION



Name Surname : Yüksel ARSLAN
Date of Birth : 08.02.1968
Phone : 0536 382 8541
E-mail : yuksel_arslan@yahoo.com

EDUCATION

High School : Naval High School
Bachelor : Naval Academy
Master Degree : Middle East Technical University

WORK EXPERIENCE

Engineering Officer: Turkish Naval Forces (1990 - 2001)
Software Engineer : Mikes Inc. (2003-2016)
Founder : SoundLabAI Inc. (2018 -)

TOPICS OF INTEREST

- Computer Networks
- Machine Learning
- Audio Signal Processing
- Software Engineering